# NMST539

# Mnohorozměrná analýza
# Multivariate Analysis

# Lecture Notes I (Classical Topics)

Ivan Mizera

*In scientific subjects, the natural remedy for dogmatism has been found in research. By temperament and training, the research worker is the antithesis of the pundit. What he is actively and constantly aware of is his ignorance, not his knowledge; the insufficiency of his concepts, of the terms and phrases in which he tries to excogitate his problems: not their final and exhaustive sufficiency. He is, therefore, usually only a good teacher for the few who wish to use their mind as a workshop, rather than a warehouse.*

*Ronald Aylmer Fisher*

## Some Basics First (Quantitative Data)

## Data matrix

The typical organization of the data in many problems in multivariate analysis is into a data matrix **Y**, in which

- rows corresponds to different objects (“cases”)

- and columns to different attributes of the data (“variables”)

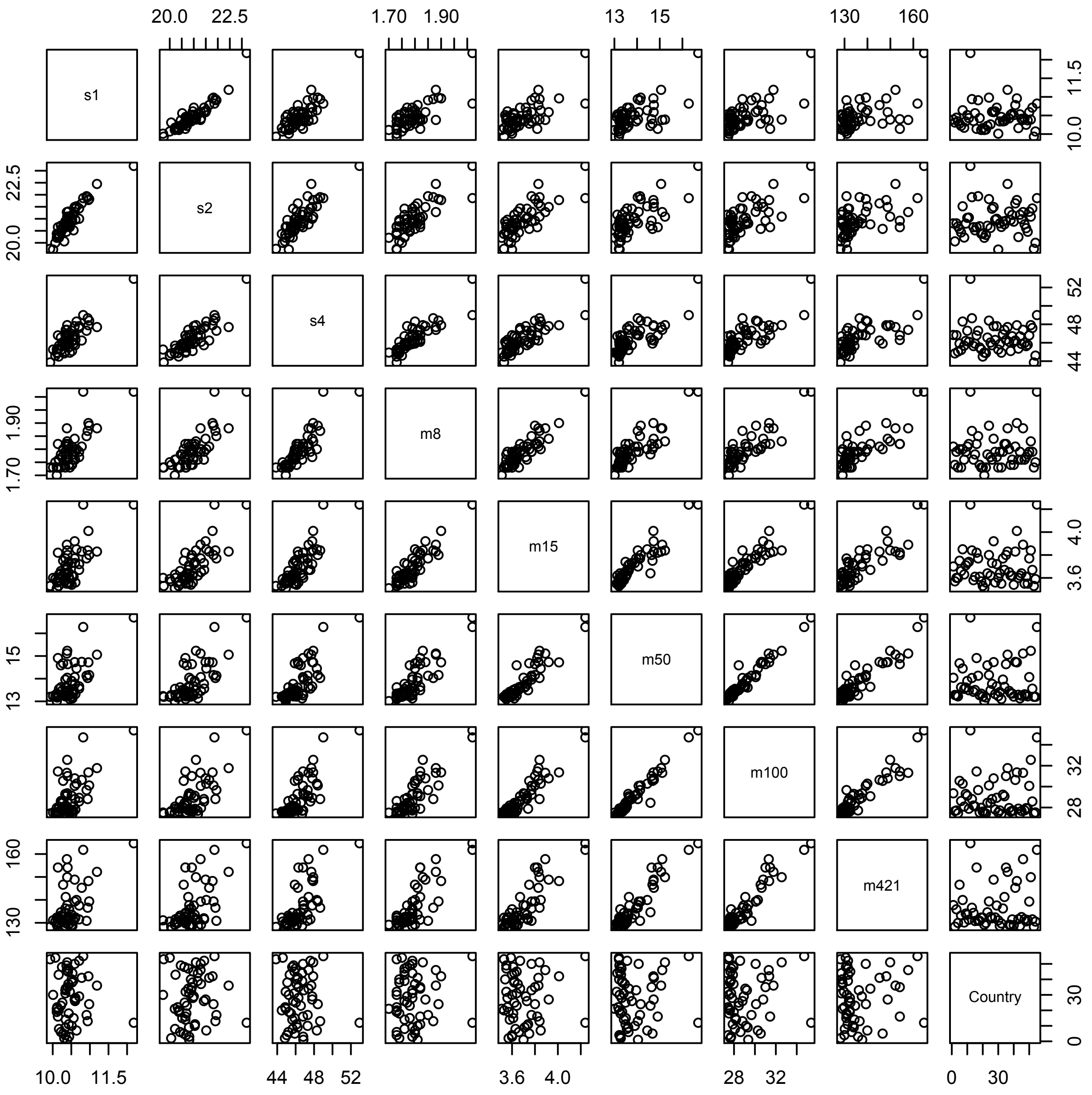
Example: Track Records for Men

The data give best achieved times by men in various countries, and various running disciplines

```
> tram
      s1    s2    s4   m8  m15   m50  m100   m421  Country
1  10.39 20.81 46.84 1.81 3.70 14.04 29.36 137.72 argentin
2  10.31 20.06 44.84 1.74 3.57 13.28 27.66 128.30 australi
3  10.44 20.81 46.82 1.79 3.60 13.26 27.72 135.90  austria
4  10.34 20.68 45.04 1.73 3.60 13.22 27.45 129.95  belgium
5  10.28 20.58 45.91 1.80 3.75 14.68 30.55 146.62  bermuda
...
```

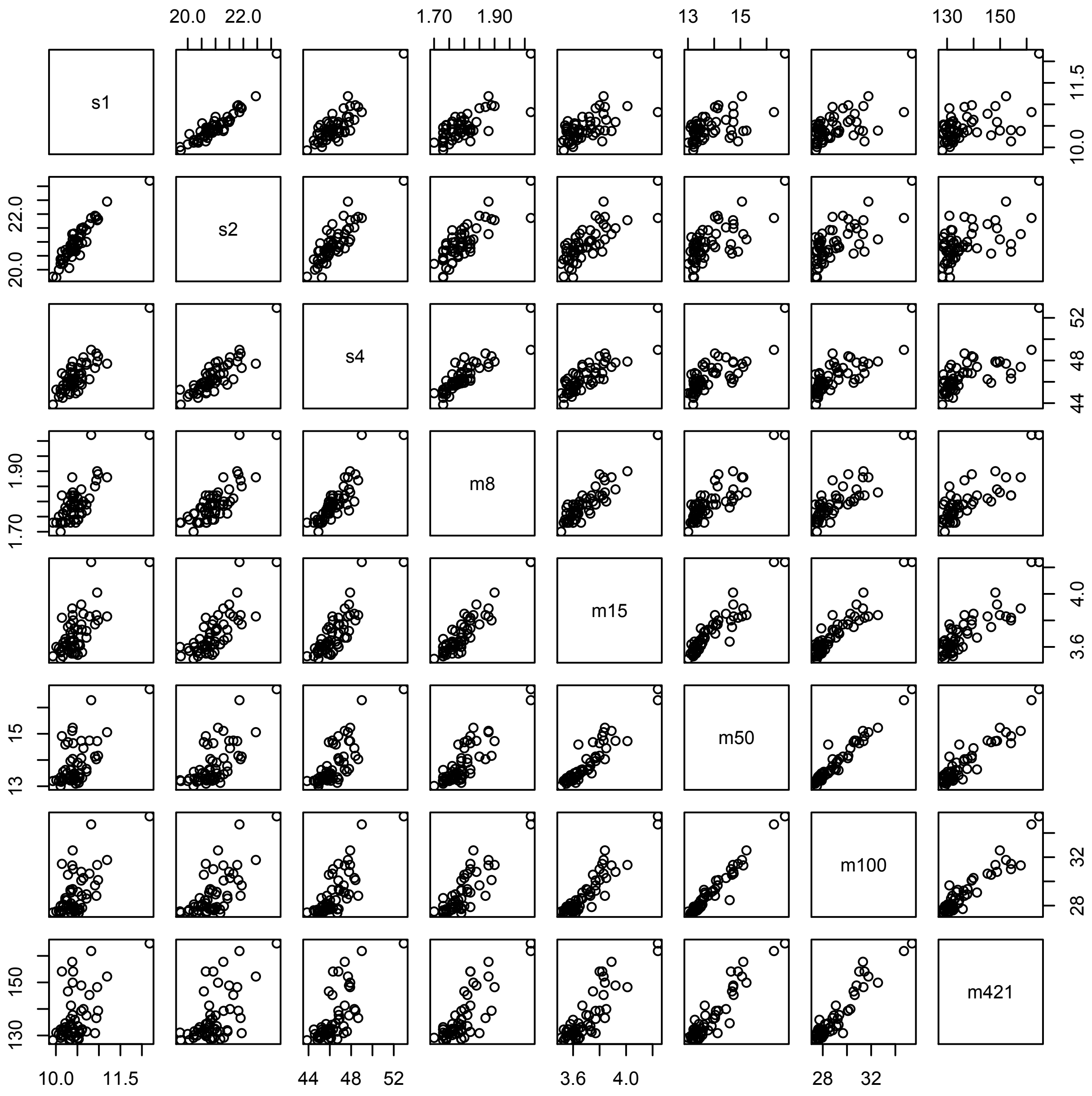
Any graphical representation for this type of data? Well, we could try (all) just all pairs of coordinate directions...

## ...but not like this...



```
> pairs(tram)
```

## ...but rather like this!



```
> pairs(tram[,1:8])
```

## Some preprocessing of the data may help too

```
> tram
     s1    s2    s4   m8  m15   m50  m100   m421  Country
1  10.39 20.81 46.84 1.81 3.70 14.04 29.36 137.72 argentin
2  10.31 20.06 44.84 1.74 3.57 13.28 27.66 128.30 australi
3  10.44 20.81 46.82 1.79 3.60 13.26 27.72 135.90  austria
...
> trackmen = tram[,1:8]
> row.names(trackmen) = tram[,9]
> trackmen
            s1    s2    s4   m8  m15   m50  m100   m421
argentin 10.39 20.81 46.84 1.81 3.70 14.04 29.36 137.72
australi 10.31 20.06 44.84 1.74 3.57 13.28 27.66 128.30
austria  10.44 20.81 46.82 1.79 3.60 13.26 27.72 135.90
...
```

## Some more preprocessing perhaps

We often want to **center** the data: subtract the (corresponding) column means

```
> trackmenc = sweep(trackmen,2,apply(trackmen,2,mean))
> apply(trackmenc,2,mean)
           s1            s2            s4             m8           m15
-3.552625e-16 1.033536e-15 2.196231e-15 -6.459578e-17 -6.459677e-17
          m50          m100          m421
 5.167584e-16 7.751770e-16 5.684247e-15
```

and also **scale** them: divide by the (corresponding) column standard deviations

```
> trackmens = sweep(trackmenc,2,apply(trackmenc,2,sd),"/")
```

If we center them and subsequently (?) scale them, we will say that we **standardize** them

(These are rather trivial modifications, unimportant from the theoretical point of view, but may play decisive rôle in practical data analysis)

## Really? (One has to check sometimes...)

```
> trackmen1 = sweep(trackmen,2,apply(trackmen,2,sd),"/")
> trackmen2 = sweep(trackmen1,2,apply(trackmen1,2,mean))
> trackmens == trackmen2
            s1    s2    s4    m8   m15   m50  m100  m421
argentin FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
...
wsamoa    TRUE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE
```

What??

```
> trackmens["wsamoa",] - trackmen2["wsamoa",]
       s1          s2           s4            m8 m15           m50
wsamoa  0 2.88658e-15 8.881784e-16 -1.332268e-15   0 -4.440892e-16 2
```

I see...

```
> round(trackmens["wsamoa",] - trackmen2["wsamoa",],digits=14)
       s1 s2 s4 m8 m15 m50 m100 m421
wsamoa  0  0  0  0   0   0    0    0
```

## Another dataset: electrodes

Various electrodes have been tried successively on arms of 16 subjects. The question whether electrodes have different characteristics or not.

```
> read.table("electro.d",header=T,row.names=1)
    E1   E2   E3  E4  E5
1  500  400   98 200 250
2  660  600  600  75 310
3  250  370  220 250 220
4   72  140  240  33  54
5  135  300  450 430  70
6   27   84  135 190 180
7  100   50   82  73  78
8  105  180   32  58  32
9   90  180  220  34  64
10 200  290  320 280 135
11  15   45   75  88  80
12 160  200  300 300 220
13 250  400   50  50  92
14 170  310  230  20 150
15  66 1000 1050 280 220
16 107   48   26  45  51
```

If there were two electrodes only, we would just compare the results on those: as they may likely for a given subject be dependent, we rather form differences and then compare those to zero: paired t-test, for instance, if we can assume normal distribution for the data

## Covariance arrays of random variables

Suppose that **Y** and **Z** are arrays consisting of random variables $y_I$ and $z_J$, where $I$ and $J$ may be more complex collections of indices (for instance, if **Z** is a matrix, then its elements may be $z_{k\ell}$. We can consider $\mathrm{Cov}(\mathbf{Y}, \mathbf{Z})$ to be the array indexed jointly by $I$ and $J$, with the elements $\mathrm{Cov}(y_I, z_J)$ - the **covariance array** of **Y** and **Z**.

If $\mathbf{Y} = \mathbf{y} = (y_1, y_2, \ldots, y_p)^\mathsf{T}$ and $\mathbf{Z} = \mathbf{z} = (z_1, z_2, \ldots, z_q)^\mathsf{T}$ are random vectors, we can conveniently represent the covariance array as a matrix. The **covariance matrix** of vectors **y** and **z** is defined to be

$$\mathrm{Cov}(\mathbf{y}, \mathbf{z}) = [\mathrm{Cov}(y_i, z_j)]$$

$$= \begin{pmatrix} \mathrm{Cov}(y_1, z_1) & \mathrm{Cov}(y_1, z_2) & \ldots & \mathrm{Cov}(y_1, z_q) \\ \mathrm{Cov}(y_2, z_1) & \mathrm{Cov}(y_2, z_2) & \ldots & \mathrm{Cov}(y_2, z_q) \\ \multicolumn{4}{c}{\dotfill} \\ \mathrm{Cov}(y_p, z_1) & \mathrm{Cov}(y_p, z_2) & \ldots & \mathrm{Cov}(y_p, z_q) \end{pmatrix}$$

where $\mathrm{Cov}(y_k, z_\ell) = \mathrm{E}\,[(y_k - \mathrm{E}(y_k))(z_\ell - \mathrm{E}(y_\ell))]$

Right from the definition we have $\mathrm{Cov}(\mathbf{z}, \mathbf{y}) = \mathrm{Cov}(\mathbf{y}, \mathbf{z})^\mathsf{T}$

Note: although covariance matrix is conveniently defined in terms of *random vectors*, it is only their *joint distribution* it depends upon

## Sample covariance matrix of two data matrices

Let $\mathbf{Y}$ and $\mathbf{Z}$ be two data matrices *with the same number of rows*. The **sample covariance matrix** of $\mathbf{Y}$ and $\mathbf{Z}$ is the matrix $\text{cov}(\mathbf{Y}, \mathbf{Z})$ whose element in the $k$-th row and $\ell$-th column is (for all $k$ and $\ell$)

$$\text{cov}(\mathbf{Y}, \mathbf{Z}) = \frac{1}{n-1} \sum_i (y_{ik} - \bar{y}_k)(z_{i\ell} - \bar{z}_\ell)$$

where $\bar{y}_k = \frac{1}{n}\sum_i y_{ij}$ and $\bar{z}_\ell = \frac{1}{n}\sum_i y_{i\ell}$ are the averages of the corresponding $k$-th and $\ell$-th columns of $\mathbf{Y}$ and $\mathbf{Z}$ (replacing the expected values from the stochastic version)

Note the division by $n-1$: while it is the most common choice, it is not universally accepted (for instance, normal distribution maximum likelihood theory may suggest division by $n$).

## Variance-covariance matrix of a random vector

Let $\mathbf{y}$ be a random vector, a collection of random variables $y_1$, $y_2$, ..., $y_p$. The **variance-covariance matrix** of $\mathbf{y}$ is defined to be

$$\mathrm{Var}(\mathbf{y}) = \mathrm{Cov}(\mathbf{y},\mathbf{y}) = [\mathrm{Cov}(y_i,y_j)]$$

$$= \begin{pmatrix} \mathrm{Cov}(y_1,y_1) & \mathrm{Cov}(y_1,y_2) & \dots & \mathrm{Cov}(y_1,y_p) \\ \mathrm{Cov}(y_2,y_1) & \mathrm{Cov}(y_2,y_2) & \dots & \mathrm{Cov}(y_2,y_p) \\ \dots & \dots & \dots & \dots \\ \mathrm{Cov}(y_p,y_1) & \mathrm{Cov}(y_p,y_2) & \dots & \mathrm{Cov}(y_p,y_p) \end{pmatrix}$$

$$= \begin{pmatrix} \mathrm{Var}(y_1) & \mathrm{Cov}(y_1,y_2) & \dots & \mathrm{Cov}(y_1,y_p) \\ \mathrm{Cov}(y_2,y_1) & \mathrm{Var}(y_2) & \dots & \mathrm{Cov}(y_2,y_p) \\ \dots & \dots & \dots & \dots \\ \mathrm{Cov}(y_p,y_1) & \mathrm{Cov}(y_p,y_2) & \dots & \mathrm{Var}(y_p) \end{pmatrix}$$

noting that $\mathrm{Cov}(y_i,y_i) = \mathrm{Var}(y_i) = \mathrm{E}\left[(y_i - \mathrm{E}(y_i))^2\right]$

Note again: the variance-covariance matrix, albeit defined in terms of random vectors, depends only on their joint distribution. We could thus speak about “variance-covariance matrix of a (multivariate) distribution” - but defining it in directly in terms of this distribution would be tedious

## Matrix notation

In the matrix notation, $\mathbf{y}$ is a vector - which we like to be a *column* one: $\mathbf{y} = (y_1, y_2, \ldots, y_p)^\top$, that is

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \cdots \\ y_p \end{pmatrix} \qquad \text{and then} \qquad \mathrm{Var}(\mathbf{y}) = \mathrm{E}[(\mathbf{y} - \mathrm{E}(\mathbf{y}))(\mathbf{y} - \mathrm{E}(\mathbf{y}))^\top]$$

In particular, when $\mathrm{E}(\mathbf{y}) = \mathbf{0}$, then $\mathrm{Var}(\mathbf{y}) = \mathrm{E}[\mathbf{y}\mathbf{y}^\top]$

This formalism gives a neat way to observe that $\mathrm{Var}(\mathbf{y})$ is always nonnegative definite (or positive semidefinite, if you wish), and also yields the formula

$$\mathrm{Var}(\mathbf{A}\mathbf{y}) = \mathbf{A}\,\mathrm{Var}(\mathbf{y})\mathbf{A}^\top$$

holding true for any (nonrandom, not necessarily square) matrix $\mathbf{A}$ that can be multiplied with $\mathbf{y}$ as indicated (that is, it has $p$ columns). Its special case is the elementary formula

$$\begin{aligned}\mathrm{Var}(a_1y_1 + a_2y_2) &= \mathrm{Var}(a_1y_1) + 2\,\mathrm{Cov}(a_1y_1, a_2y_2) + \mathrm{Var}(a_2y_2) \\ &= a_1^2\,\mathrm{Var}(y_1) + 2a_1a_2\,\mathrm{Cov}(y_1, y_2) + a_2^2\,\mathrm{Var}(y_2)\end{aligned}$$

## Sample variance-covariance matrix

Analogously, the **sample variance-covariance matrix** of a data matrix $\mathbf{Y}$ is the sample covariance matrix of $\mathbf{Y}$ and $\mathbf{Y}$,

$\mathbf{S}_{\mathbf{Y}} = \mathrm{var}(\mathbf{Y}) = \mathrm{cov}(\mathbf{Y}, \mathbf{Y})$

Right from the definition it follows that this matrix contains the sample covariance of the $k$-th and $\ell$-th columns of $\mathbf{Y}$ in its $k$-th row and $\ell$-th column (and, for this matter, also in its $\ell$-th row and $k$-th column)

$$\mathrm{var}(\mathbf{Y}) = \begin{pmatrix} s_{11} & s_{12} & \dots & s_{1p} \\ s_{21} & s_{22} & \dots & s_{2p} \\ \dots & \dots & \dots & \dots \\ s_{p1} & s_{p2} & \dots & s_{pp} \end{pmatrix} = \begin{pmatrix} s_1^2 & s_{12} & \dots & s_{1p} \\ s_{21} & s_2^2 & \dots & s_{2p} \\ \dots & \dots & \dots & \dots \\ s_{p1} & s_{p2} & \dots & s_p^2 \end{pmatrix}$$

Thus, the matrix is symmetric, and also nonnegative definite (some say: positive semidefinite) - which can be best seen from its matrix expression

## The compact matrix notation

The centered data matrix is $\tilde{\mathbf{Y}} = \mathbf{Y} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top\mathbf{Y}$.

The (sample) **variance-covariance matrix** of $\mathbf{Y}$ is

$$\begin{aligned}
\mathbf{S}_{\mathbf{Y}} &= \frac{1}{n-1}\tilde{\mathbf{Y}}^\top\tilde{\mathbf{Y}} = \frac{1}{n-1}\left(\mathbf{Y} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top\mathbf{Y}\right)^\top\left(\mathbf{Y} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top\mathbf{Y}\right) \\
&= \frac{1}{n-1}\mathbf{Y}^\top\left(\mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top\right)^\top\left(\mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top\right)\mathbf{Y} \\
&= \frac{1}{n-1}\mathbf{Y}^\top\left(\mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top\right)\left(\mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top\right)\mathbf{Y} = \frac{1}{n-1}\mathbf{Y}^\top\left(\mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top\right)\mathbf{Y}
\end{aligned}$$

The matrix $\mathbf{H} = \mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$ is symmetric and idempotent

- which means that $\mathbf{H}\mathbf{H} = \mathbf{H}$

Idempotence immediately implies that the matrix is nonnegative definite - and its rank is equal to its trace: $n-1$

## Sample variance-covariance matrix: transformation theory

For the stochastic version of the variance-covariance matrix, we have the transformation formula, for any nonrandom $\mathbf{A}$

The transformation theory, including formulation and proof of the analog of the above formula, for the sample variance-covariance matrices, working with the (non-random) data matrix $\mathbf{Y}$ instead of the random vector $\mathbf{y}$, and considering *the same* (non-random) $\mathbf{A}$, works with[1] ...

$\mathbf{YA}$ instead of $\mathbf{Ay}$; the corresponding transformation formula is then

$$\mathrm{Var}(\mathbf{YA}) = \mathbf{A}^\top \mathrm{Var}(\mathbf{Y})\mathbf{A}$$

[1] Problem 13

## Correlation matrix

The **(sample) correlation matrix** of $\mathbf{Y}$ is

$$\mathrm{cor}(\mathbf{Y}) = \begin{pmatrix} \rho_{11} & \rho_{12} & \dots & \rho_{1p} \\ \rho_{21} & \rho_{22} & \dots & \rho_{2p} \\ \dots & \dots & \dots & \dots \\ \rho_{p1} & \rho_{p2} & \dots & \rho_{pp} \end{pmatrix} = \begin{pmatrix} 1 & \rho_{12} & \dots & \rho_{1p} \\ \rho_{21} & 1 & \dots & \rho_{2p} \\ \dots & \dots & \dots & \dots \\ \rho_{p1} & \rho_{p2} & \dots & 1 \end{pmatrix}$$

where $\rho_{k\ell}$ is the sample correlation coefficient

$$\rho_{k\ell} = \frac{\sum_{i=1}^{n} (y_{ik} - \bar{y}_k)(y_{i\ell} - \bar{y}_\ell)}{\sqrt{\sum_{i=1}^{n} (y_{ik} - \bar{y}_k)^2 \sum_{i=1}^{n} (y_{i\ell} - \bar{y}_\ell)^2}}$$

The sample variance-covariance matrix computed out of the scaled data is the correlation matrix[2]

[2] Problem 15

## Correlation matrix of a random vector

The (*stochastic*) **correlation matrix**, $\mathrm{Cor}(\mathbf{y})$, of a random vector $\mathbf{y}$ is defined analogously, only the $\rho_{k\ell}$'s are replaced by stochastic correlation coefficients

$$\frac{\mathrm{Cov}(y_k, y_\ell)}{\sqrt{\mathrm{Var}(y_k)\,\mathrm{Var}(y_\ell)}}$$

## Notes on the terminology

Especially in the applied literature, the adjective "sample" is often omitted; the type of the variance-covariance or covariance matrices should be decided from the context. As the applications are always to data, not to random variables, "variance matrix" or "covariance matrix" in applied literature mostly refer to the sample versions; should they pertain to random vectors, it might be emphasized by an adjective "stochastic"

In statistical theory, on the other hand, the sample versions are viewed as estimates of the "underlying" stochastic ones, using the "principle of analogy" the data are considered rowwise sampled from a common distribution; see below. More important than this motivation is to remember the result: (co)variances $\mathrm{Cov}(y_k, y_\ell)$ are replaced by the corresponding sample (co)variances

Also, although we stick here to the "variance-covariance" terminology, in literature (especially the applied one) they often say just shortly *variance matrix* (or sometimes even *covariance matrix*), for any of the variants (sample or stochastic) considered above

## Seeing the Data: A Word on Projections

## Dimensionality problem

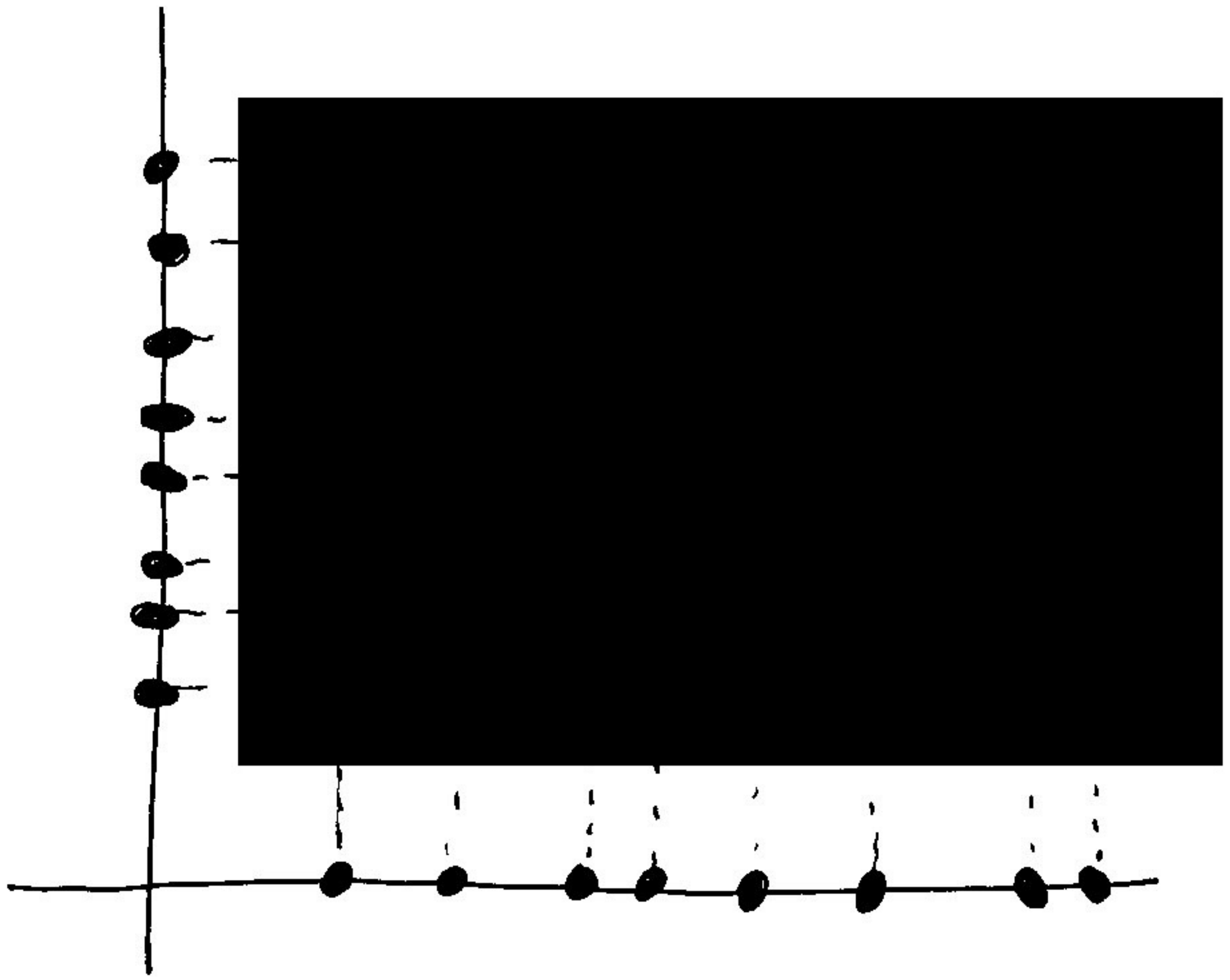
We live in dim 3 and see in dim 2

To understand this more:

    we may imagine that we live in dim 2 and see in dim 1

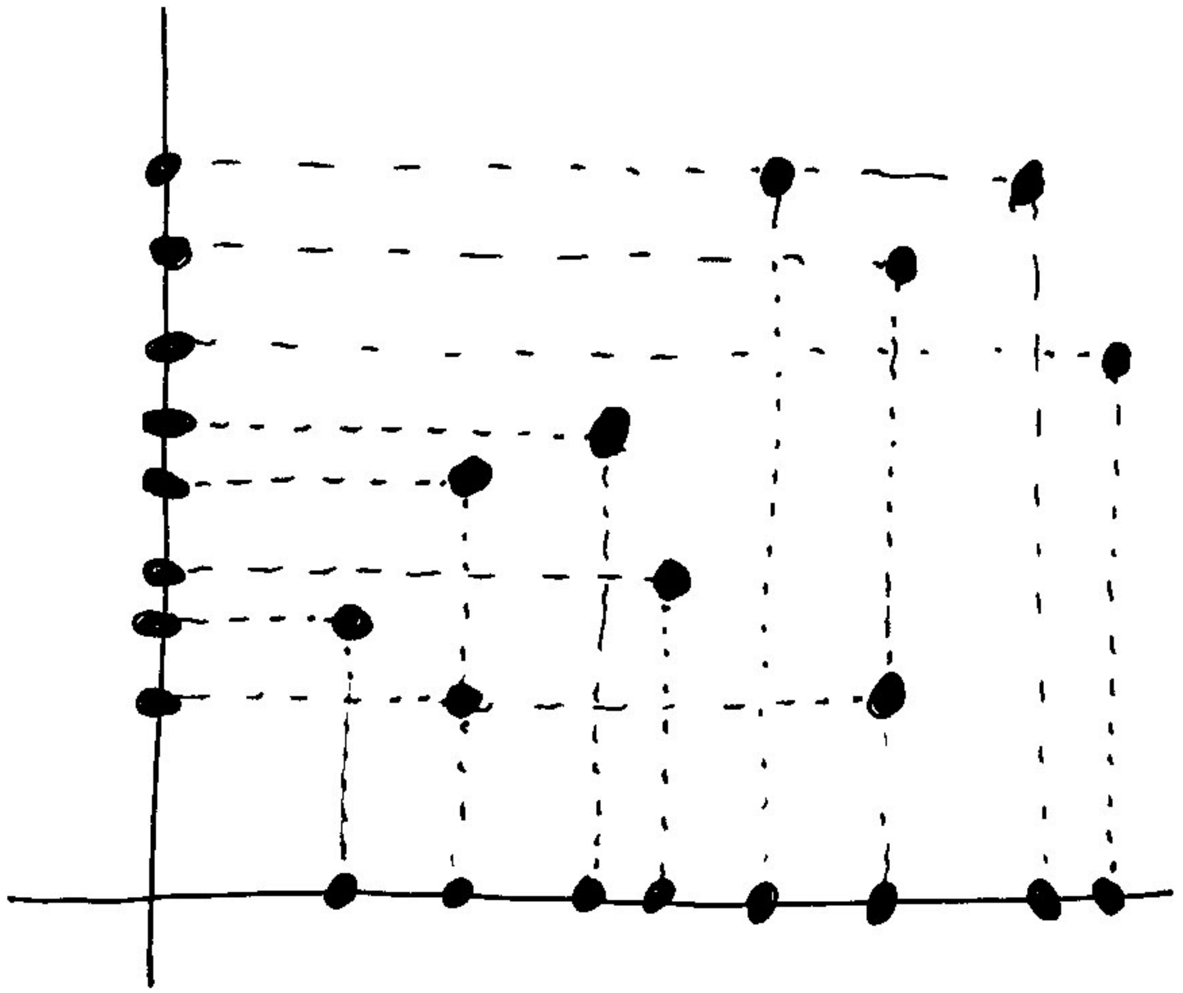
(In multivariate analysis we still can see maximally in dim 2
- but live in dim 10, 25, 300, ...)

## Projections on coordinate axes

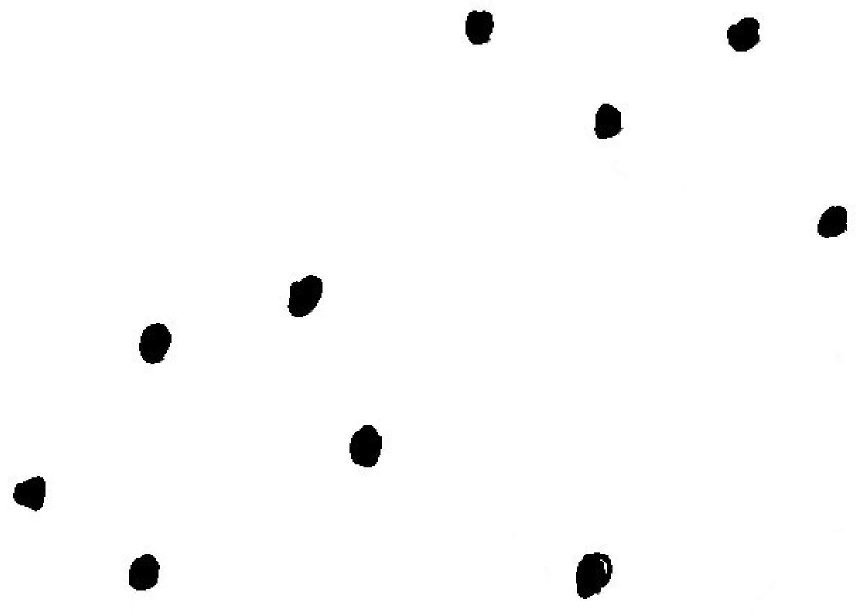


Are these views interesting?

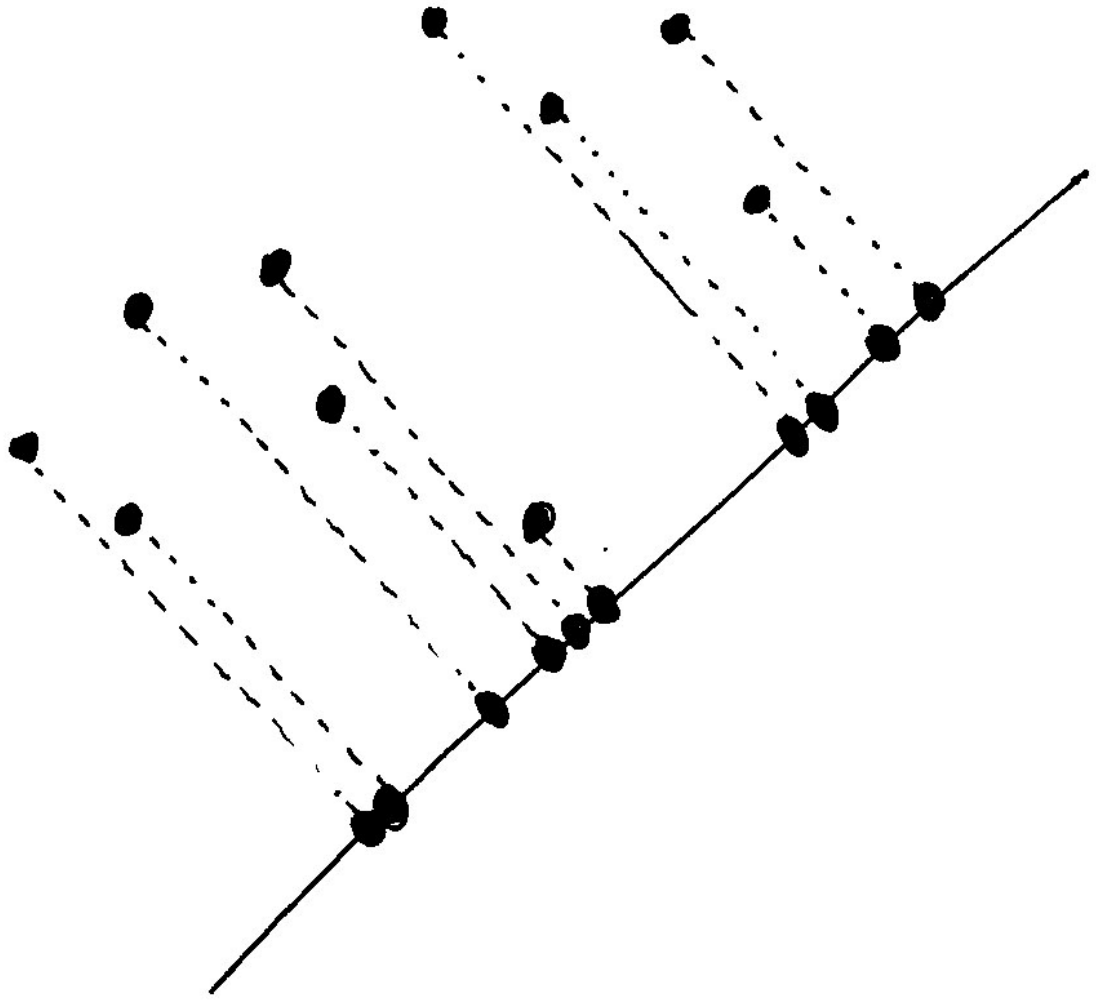
## The data revealed



## Even more revealed



## Interesting projection



How to find it?

## General theory

A **projection** is a specific mapping (and its result) into a (lower-dimensional) subspace: if $\Omega$ is that subspace, then given $\mathbf{x}$, we seek the element $\mathbf{y}$ that minimizes $\|\mathbf{y}-\mathbf{x}\|$ over $\mathbf{y} \in \Omega$

For the Euclidean distance/norm, a neat solution is obtained via the **orthogonal complement** $\Omega^{\perp} = \{\mathbf{u} : \mathbf{u}^{\mathsf{T}}\mathbf{v} = 0 \text{ for all } \mathbf{v} \in \Omega\}$:

if $\mathbf{x}$ can be written as $\mathbf{x} = \mathbf{y} + \mathbf{z}$, where $\mathbf{y}$ is in $\Omega$ and $\mathbf{z}$ in $\Omega^{\perp}$, then $\mathbf{y}$ is the projection of $\mathbf{x}$ on $\Omega$

The fact that $\mathbf{y}$ is the projection follows from the Pythagoras theorem - similarly as the fact that no projection increases norms/lengths

The projection onto $\Omega$ as a mapping maps $\mathbf{x}$ into the $\mathbf{y}$ as defined above. It is a linear operator, the **projection operator**, which is represented by a matrix $\mathbf{P}_{\Omega}$; this matrix is symmetric and **idempotent**: $\mathbf{P}_{\Omega}\mathbf{P}_{\Omega} = \mathbf{P}_{\Omega}$

## Subspaces via generators

If the subspace $\Omega$ to be projected to is given via generating vectors that form columns of a matrix $\mathbf{A}$ - that is, $\Omega = \mathrm{Im}(\mathbf{A})$, sometimes denoted also $\mathcal{M}(\mathbf{A})$, where

$$\mathrm{Im}(\mathbf{A}) = \{\mathbf{x} : \mathbf{x} = \mathbf{A}\mathbf{y} \text{ for some } \mathbf{y}\}$$

then the matrix of the projection to $\mathrm{Im}(\mathbf{A})$

$$\mathbf{P}_{\mathrm{Im}(\mathbf{A})} = \mathbf{A}(\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top$$

Obviously, this works if the matrix $\mathbf{A}^\top\mathbf{A}$ is invertible, which happens if and only if its columns, the generating vectors, are linearly independent (otherwise, one has to work with pseudo-inverse matrices instead)

## Pictorial projection

In multivariate data analysis, under “projection” they often mean “pictorial projection” - especially when it results in a subspace $\Omega$ with dimension $p$ equal 1 and 2. Projection, as defined above, ends up in a subspace: the projected vectors are in this subspace, which is a subset of the original vector space

Pictorial projection is what we would see if we identified $\Omega$ with a ($p$-dimensional) computer screen. This means that the projected vectors are expressed in terms of the $p$-dimensional standard basis. The simplest way of achieving this is to express the projected vectors in the coördinates given by the generating vectors, columns of the matrix A considered above; this works best if the generating vectors form an orthonormal basis

## Pictorial projection: the technology

So, suppose that we are projecting on the subspace generated by the the columns of a matrix $\mathbf{A}$, which are orthogonal and their norm is 1. In such case, $\mathbf{A}^\top\mathbf{A} = \mathbf{I}$, and the corresponding projecting matrix (as considered above) is

$$\mathbf{P}_{\mathrm{Im}(\mathbf{A})} = \mathbf{A}(\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top = \mathbf{A}\mathbf{A}^\top$$

While the projection of a vector $\mathbf{x}$ on the space generated by the columns of $\mathbf{A}$ is thus $\mathbf{A}\mathbf{A}^\top\mathbf{x}$, its pictorial projection, its coördinates in terms of the basis given by the columns of $\mathbf{A}$, is $\mathbf{A}^\top\mathbf{x}$

Compared to the projection matrix $\mathbf{A}\mathbf{A}^\top$, the matrix of the linear mapping corresponding to the pictorial projection (when $\mathbf{A}$ has orthonormal columns) is

$$\mathbf{A}^\top\mathbf{P}_{\mathrm{Im}(\mathbf{A})} = \mathbf{A}^\top\mathbf{A}\mathbf{A}^\top = \mathbf{A}^\top$$

## What you see is what you get?

Pictorial projection into 2-dimensional subspace is supposed to give what you see in when you look at the projected vectors within that 2-dimensional subspace - what you see on the computer screen if you project "in that direction"

Thus, if the columns of $\mathbf{A}$ are orthogonal, but their norm is not necessarily 1, the pictorial projection is often considered with respect to their normalizations - that is, if $\mathbf{a}$ is a generating vector, the pictorial projection coördinates are considered with respect to $\mathbf{a}/\|\mathbf{a}\|$ rather than $\mathbf{a}$ itself

Also, it should be kept in mind that R (as well as other software) often automatically (unless told not to do so) rescales the coordinates so that the resulting graph nicely fits the usual picture frame. So you may not see the angles and distances right on a default picture

Interestingly, obtaining an automatically equiscaled plot is not that straightforward as, for example, in MATLAB (`axis equal`) but one has rather to use a function `eqscplot()` from the library `MASS`

## Pictorial projections: dimension 2 to dimension 1

Into the x coordinate - the direction of $\begin{pmatrix}1\\0\end{pmatrix}$: $\begin{pmatrix}x_1\\x_2\end{pmatrix} \mapsto x_1$

Into the y coordinate - the direction of $\begin{pmatrix}0\\1\end{pmatrix}$: $\begin{pmatrix}x_1\\x_2\end{pmatrix} \mapsto x_2$

Into the direction of $\begin{pmatrix}1\\1\end{pmatrix}$: $\begin{pmatrix}x_1\\x_2\end{pmatrix} \mapsto x_1 + x_2$ (???)

... not really! we want to adjust the scale!

Projection into the direction of $\begin{pmatrix}a_1\\a_2\end{pmatrix}$ - use inner product:

$\begin{pmatrix}x_1\\x_2\end{pmatrix} \mapsto a_1x_1 + a_2x_2$, provided the length of $\left\|\begin{pmatrix}a_1\\a_2\end{pmatrix}\right\| = 1$

$\left\|\begin{pmatrix}a_1\\a_2\end{pmatrix}\right\| = 1$ means $\sqrt{a_1^2 + a_2^2} = 1$, equivalent to $a_1^2 + a_2^2 = 1$

## Compact matrix notation, general dimension

The pictorial projection of $\mathbf{x}$ into the direction of $\mathbf{a}$ is:

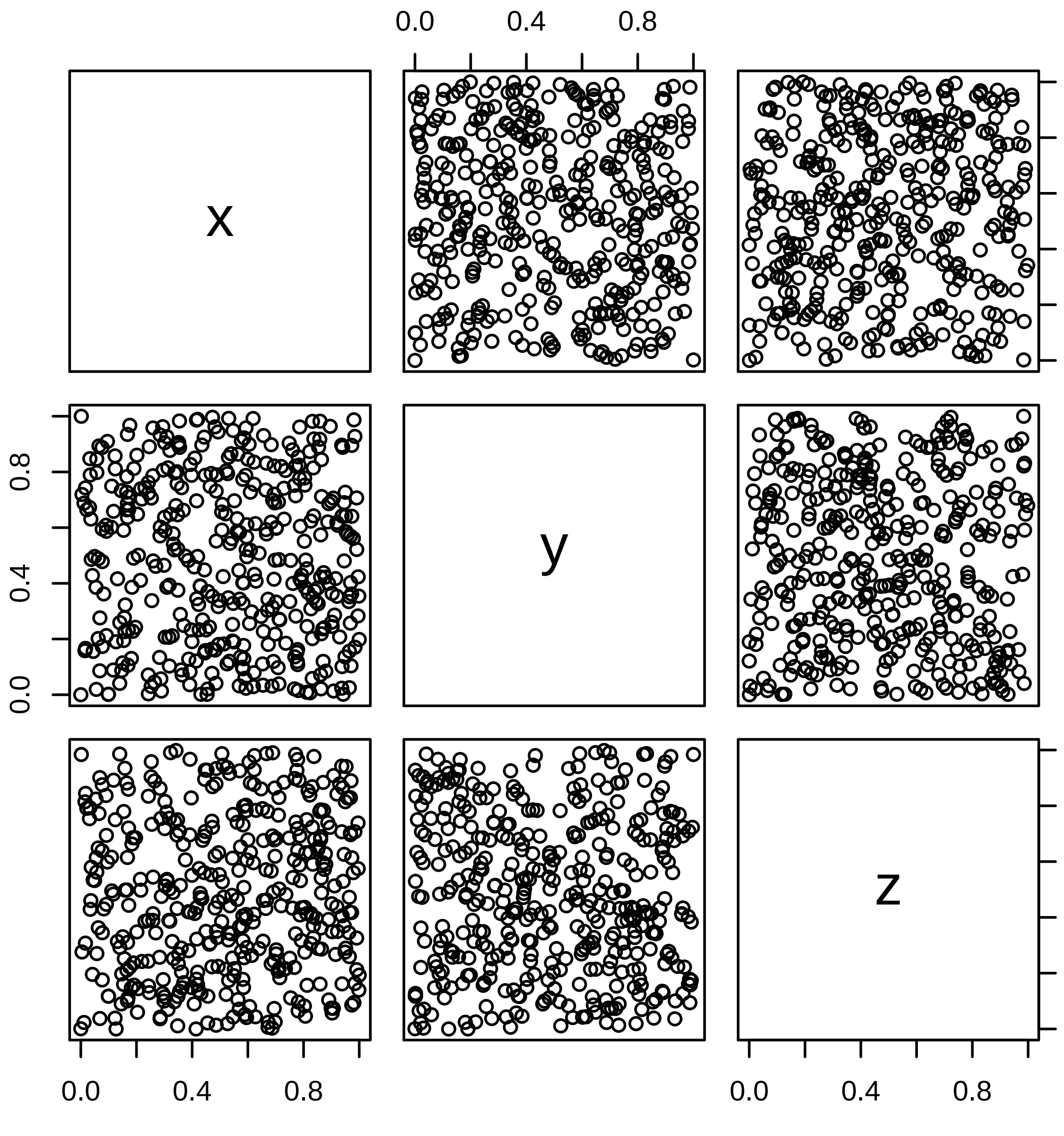
$$\mathbf{x} \mapsto \mathbf{a}^\top \mathbf{x} \qquad \text{if } \|\mathbf{a}\| = 1$$

$$\mathbf{x} \mapsto \mathbf{a}_0^\top \mathbf{x} \quad \text{where } \mathbf{a}_0 = \frac{\mathbf{a}}{\|\mathbf{a}\|} \qquad \text{in the general case}$$

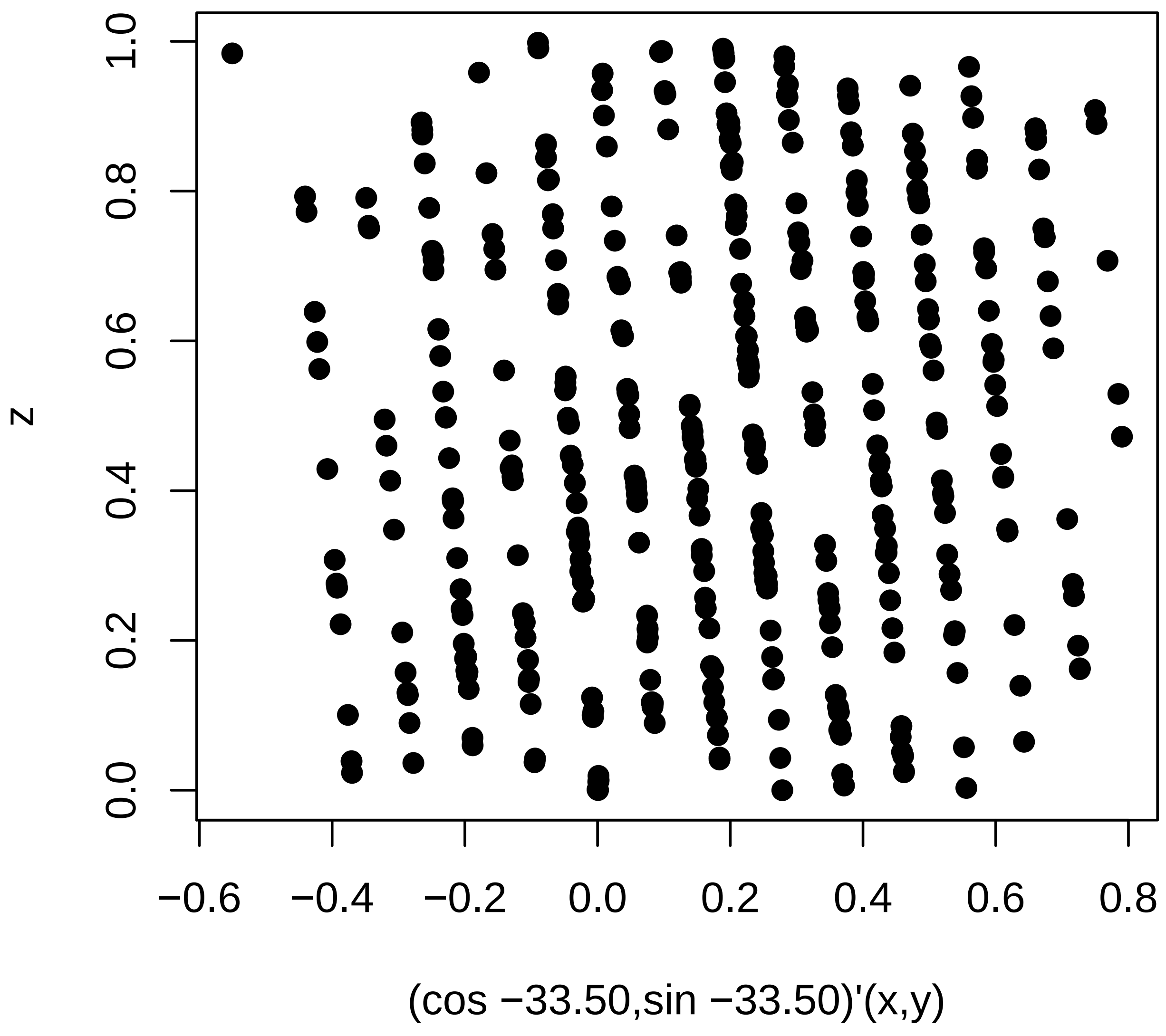
For instance, the pictorial projection into the direction of $\begin{pmatrix}1\\1\end{pmatrix}$ is

$$\begin{pmatrix}x_1\\x_2\end{pmatrix} \mapsto \frac{x_1 + x_2}{\sqrt{2}}$$

# Example: a random generator...



## ...and a lousy one!



## How was it done: the R code

```
> data(randu)
> pairs(randu)
> plot(as.matrix(randu) %*%
+ c(cos(pi*(-33.50)/180),sin(pi*(-33.50)/180),0),
+ as.matrix(randu) %*% c(0,0,1),
+ xlab="(cos -33.50,sin -33.50)'(x,y)", ylab="z", pch=16)
```

Is there an (automatic) way to find interesting projections?

## Digression: Qualitative (Categorical) Data

## Qualitative - discrete variables

Can arise by themselves, or be created by `cut()` from quantitative. If we have data recorded by items, we can use `table()` to create the cross-tabulated form. Otherwise, data with qualitative variables often come already in a tabular form.

Example: Hair and eye color and sex in 592 statistics students

```
> HairEyeColor
, , Sex = Male
       Eye
Hair    Brown Blue Hazel Green
  Black    32   11    10     3
  Brown    53   50    25    15
  Red      10   10     7     7
  Blond     3   30     5     8
, , Sex = Female
       Eye
Hair    Brown Blue Hazel Green
  Black    36    9     5     2
  Brown    66   34    29    14
  Red      16    7     7     7
  Blond     4   64     5     8
> class(HairEyeColor)
[1] "table"
```

## As data frame

The inverse of `table()` is

```
> hair=as.data.frame(HairEyeColor)
> hair
    Hair   Eye    Sex Freq
1  Black Brown   Male   32
2  Brown Brown   Male   53
3    Red Brown   Male   10
4  Blond Brown   Male    3
5  Black  Blue   Male   11
6  Brown  Blue   Male   50
7    Red  Blue   Male   10
8  Blond  Blue   Male   30
9  Black Hazel   Male   10
10 Brown Hazel   Male   25
11   Red Hazel   Male    7
12 Blond Hazel   Male    5
13 Black Green   Male    3
14 Brown Green   Male   15
15   Red Green   Male    7
16 Blond Green   Male    8
```

```
17 Black Brown Female   36
18 Brown Brown Female   66
19   Red Brown Female   16
20 Blond Brown Female    4
21 Black  Blue Female    9
22 Brown  Blue Female   34
23   Red  Blue Female    7
24 Blond  Blue Female   64
25 Black Hazel Female    5
26 Brown Hazel Female   29
27   Red Hazel Female    7
28 Blond Hazel Female    5
29 Black Green Female    2
30 Brown Green Female   14
31   Red Green Female    7
32 Blond Green Female    8
```

## Conversion back?

Not `table()`, but `xtabs()`.

```
> xtabs(Freq~Hair+Eye+Sex,data=hair) # xtabs(Freq~.,data=hair)
, , Sex = Male
       Eye
Hair    Brown Blue Hazel Green
  Black    32   11    10     3
  Brown    53   50    25    15
  Red      10   10     7     7
  Blond     3   30     5     8
, , Sex = Female
       Eye
Hair    Brown Blue Hazel Green
  Black    36    9     5     2
  Brown    66   34    29    14
  Red      16    7     7     7
  Blond     4   64     5     8
```

We may need (?) to declare the right class: `class(.)="table"`.

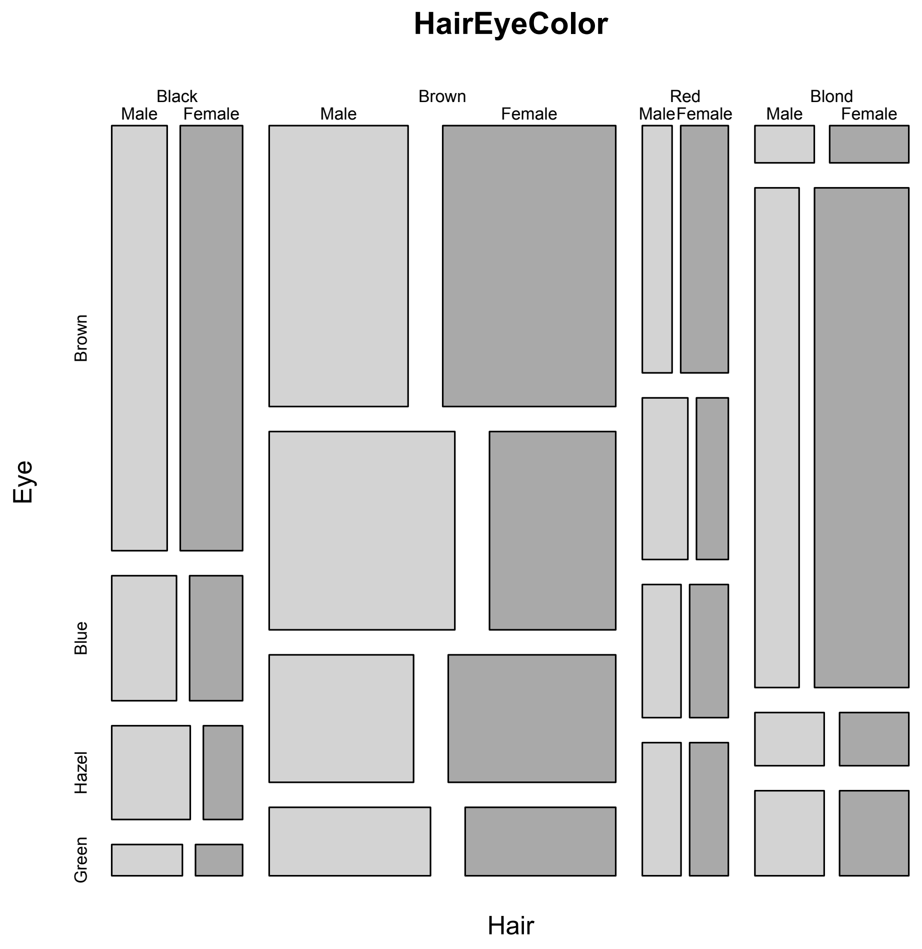
## But also “something like” the original data

```
> hairdat=hair[rep(row.names(hair),hair$Freq),1:3]
> hairdat
       Hair   Eye    Sex
1     Black Brown   Male
1.1   Black Brown   Male
1.2   Black Brown   Male
...
32.6  Blond Green Female
32.7  Blond Green Female
> table(hairdat)
, , Sex = Male

       Eye
Hair    Brown Blue Hazel Green
  Black    32   11    10     3
...
  Red      16    7     7     7
  Blond     4   64     5     8
```

## Plotting

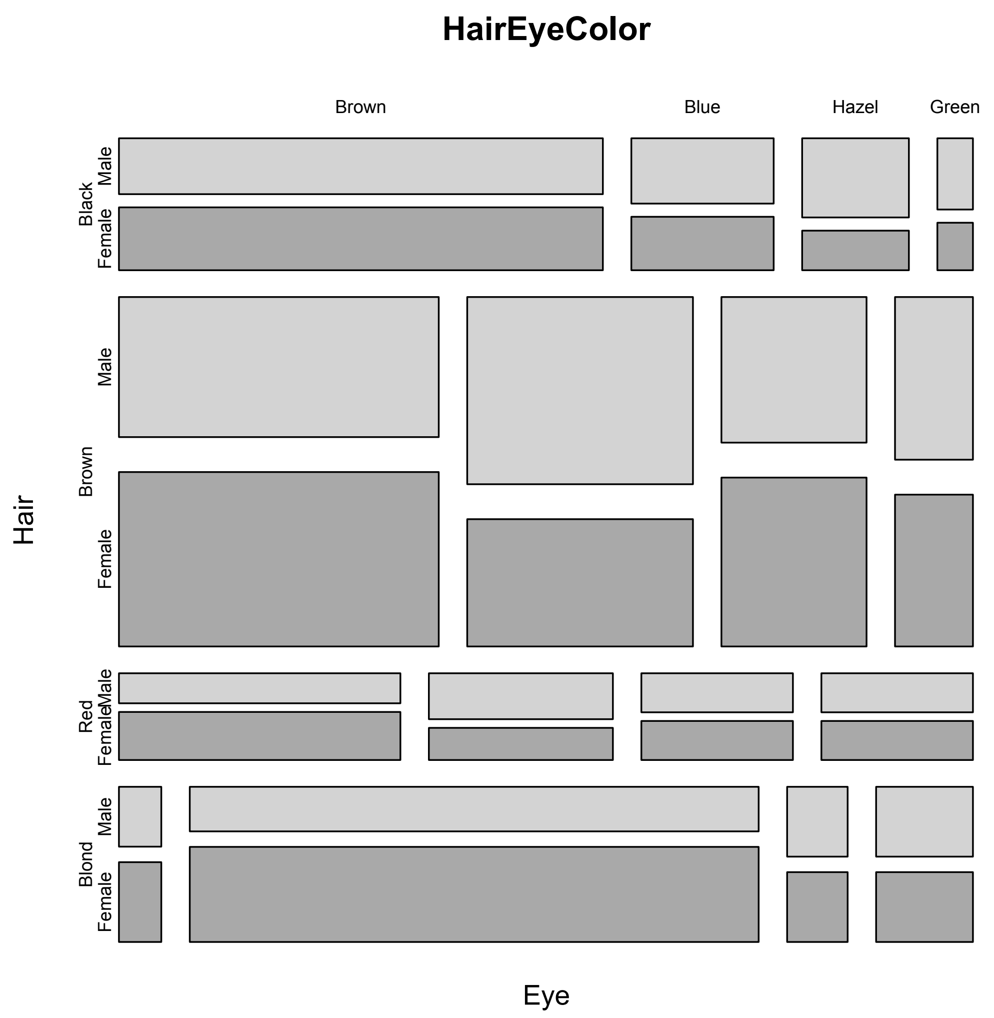
As pie charts, we leave also the “3D histograms” to programs like Excel. A real statistician uses something else: for instance

```
> mosaicplot(HairEyeColor, color=c("lightgray","darkgray"))
```



# We can control it

```
> mosaicplot(HairEyeColor,
+ color=c("lightgray","darkgray"), dir=c("h","v","h"))
```



## How about eye and hair color aggregated?

That is, among “statisticiens et statisticiennes” ...

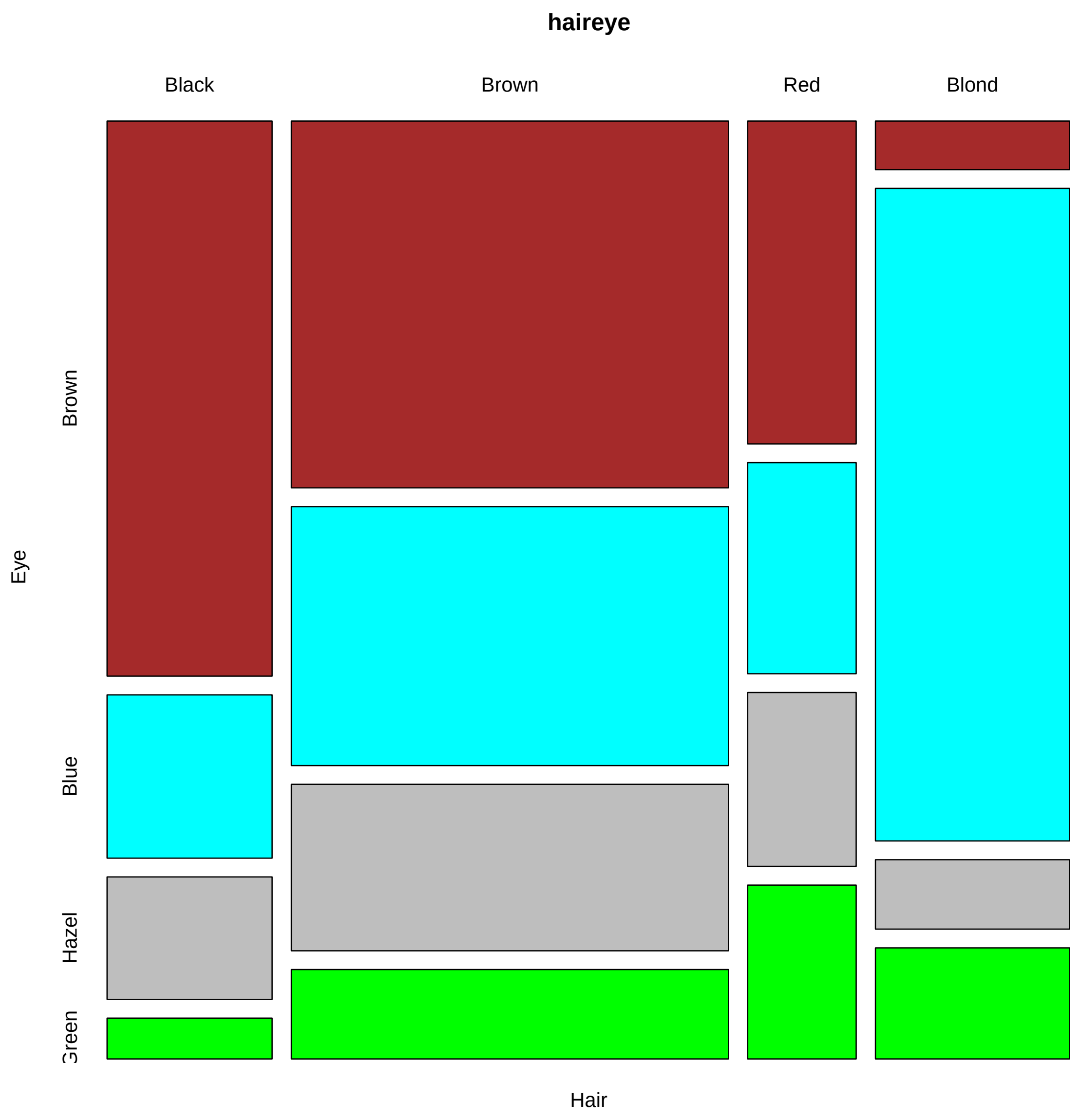
```
> haireye=apply(HairEyeColor,1:2,sum)
> haireye
       Eye
Hair     Brown Blue Hazel Green
  Black     68   20    15     5
  Brown    119   84    54    29
  Red       26   17    14    14
  Blond      7   94    10    16
> mosaicplot(haireye, color=c("brown","blue","grey","green"))
```

And give it nice colors...

Try also:

```
> mosaicplot(haireye,
+ color=c("brown","blue","grey","green"),cex=1)
```

# In color



What else we can do?

## Log-linear models

...as a special case of `glm()`. For frequency data, the distribution is Poisson; the link is the logarithm function. That is, we construct a linear model where the response are logarithms of *expected* frequences.

We see why we needed to convert the table to the data frame:

```
> hairglm=glm(Freq~Hair*Eye*Sex,family=poisson, data=hair)
> anova(hairglm,test="Chisq")
            Df Deviance Resid. Df Resid. Dev Pr(>Chi)
NULL                           31     475.12
Hair         3  165.592        28     309.53  < 2e-16 ***
Eye          3  141.272        25     168.25  < 2e-16 ***
Sex          1    1.954        24     166.30  0.16218
Hair:Eye     9  146.444        15      19.86  < 2e-16 ***
Hair:Sex     3    8.093        12      11.76  0.04413 *
Eye:Sex      3    5.002         9       6.76  0.17162
Hair:Eye:Sex 9    6.761         0       0.00  0.66196
```

Try also `summary(hairglm)`, to see why we prefer `anova(hairglm)` here. On the other hand, the problem of `anova()` is that the p-values are for incremental composite hypotheses.

## Incremental `anova`

```
> hairtwo=glm(Freq~(Hair+Eye+Sex)^2,data=hair,family=poisson)
> hairsp=glm(Freq~Hair+Eye+Sex+Hair:Eye+Hair:Sex,data=hair,family=poisson)
> anova(hairtwo,hairglm,test="Chisq")
...
Model 1: Freq ~ (Hair + Eye + Sex)^2
Model 2: Freq ~ Hair * Eye * Sex
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1         9     6.7613
2         0     0.0000  9   6.7613    0.662
> anova(hairsp,hairglm,test="Chisq")
...
Model 1: Freq ~ Hair + Eye + Sex + Hair:Eye + Hair:Sex
Model 2: Freq ~ Hair * Eye * Sex
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1        12     11.764
2         0      0.000 12   11.764   0.4648
> anova(hairsp,hairtwo,test="Chisq")
...
Model 1: Freq ~ Hair + Eye + Sex + Hair:Eye + Hair:Sex
Model 2: Freq ~ (Hair + Eye + Sex)^2
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1        12    11.7637
2         9     6.7613  3   5.0025   0.1716
```

## Also

```
> hairind=glm(Freq~Hair+Eye+Sex,family=poisson, data=hair)
> hairnosex=glm(Freq~Hair*Eye+Sex,family=poisson, data=hair)
> anova(hairnosex,hairglm,test="Chisq")
...
Model 1: Freq ~ Hair * Eye + Sex
Model 2: Freq ~ Hair * Eye * Sex
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1        15     19.857
2         0      0.000 15   19.857   0.1775
> anova(hairind,hairglm,test="Chisq")
...
Model 1: Freq ~ Hair + Eye + Sex
Model 2: Freq ~ Hair * Eye * Sex
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1        24      166.3
2         0        0.0 24    166.3 < 2.2e-16 ***
> anova(hairind,hairnosex,test="Chisq")
...
Model 1: Freq ~ Hair + Eye + Sex
Model 2: Freq ~ Hair * Eye + Sex
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1        24    166.300
2        15     19.857  9   146.44 < 2.2e-16 ***
```

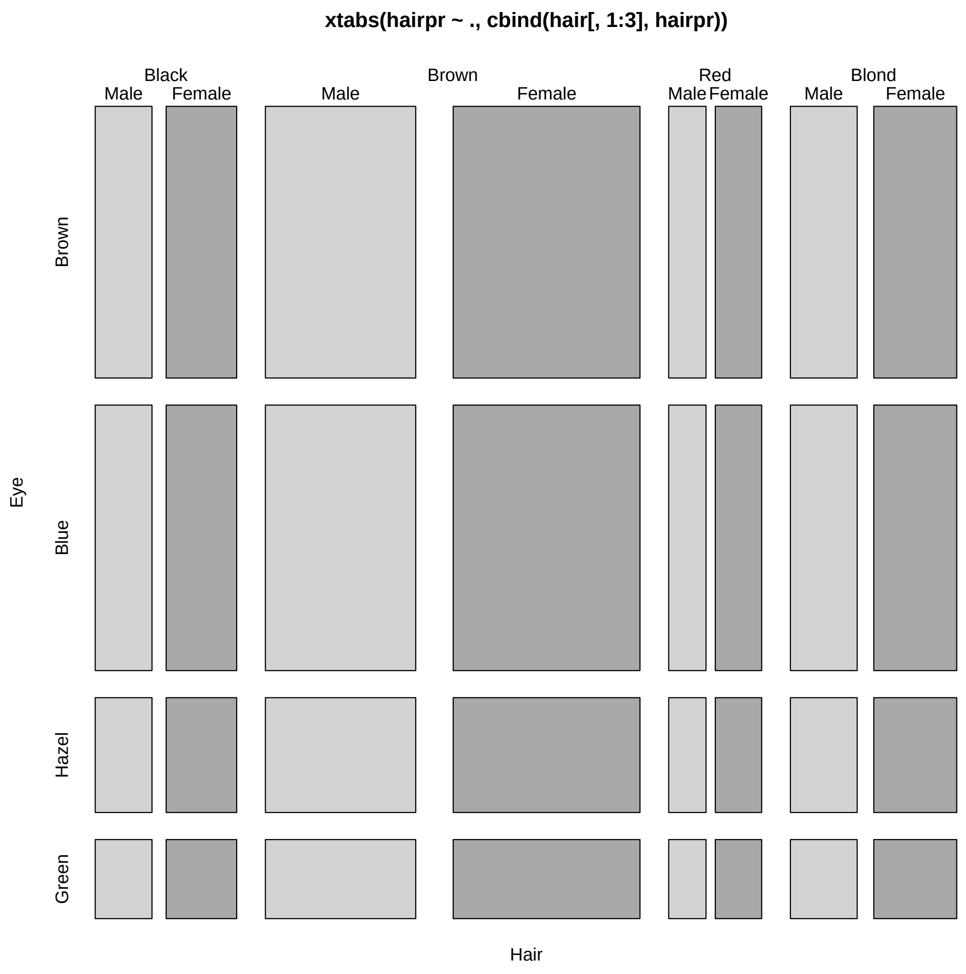
## Predicted frequences under independence

Let us take predicted values under the additive model—which in the context of qualitative data means independence:all three variables, Sex, Color of Hair, and Color of Eyes, are independent.

```
> hairpr=predict(hairind,type="response")
> xtabs(hairpr~.,cbind(hair[,1:3],hairpr))
, , Sex = Male

       Eye
Hair        Brown      Blue     Hazel     Green
  Black 18.915038 18.485151  7.995903  5.502557
  Brown 50.089824 48.951419 21.174335 14.571585
  Red   12.434886 12.152275  5.256566  3.617421
  Blond 22.242684 21.737168  9.402589  6.470599

, , Sex = Female

       Eye
Hair        Brown      Blue     Hazel     Green
  Black 21.220097 20.737822  8.970314  6.173119
  Brown 56.193960 54.916825 23.754719 16.347334
  Red   13.950249 13.633198  5.897151  4.058254
  Blond 24.953262 24.386142 10.548424  7.259131
```

## A picture?

```
> mosaicplot(xtabs(hairpr~.,cbind(hair[,1:3],hairpr)),
+ color=c("lightgray","darkgray"))
```



Not that interesting...

## We would rather like to see residuals

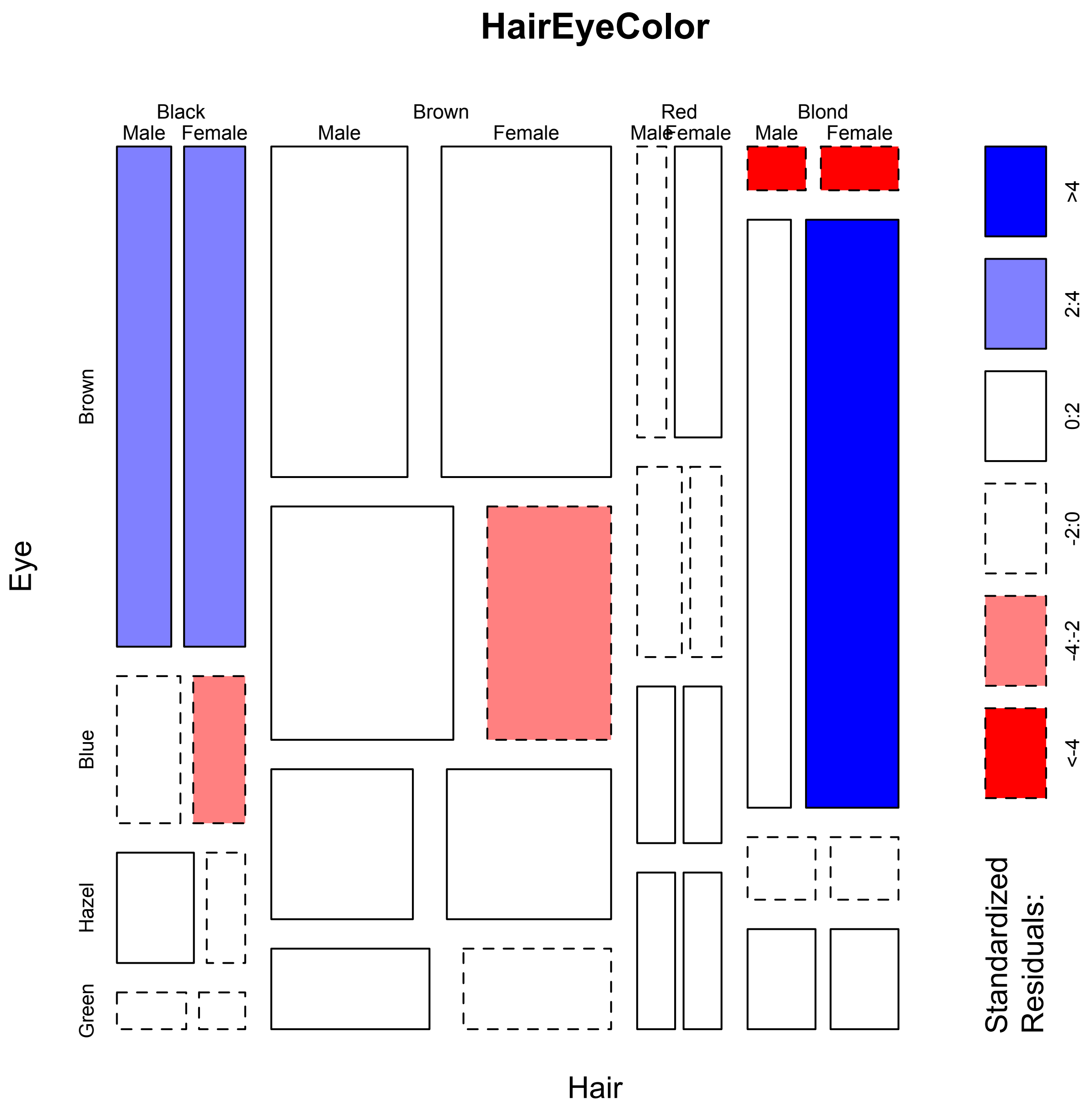
```
> hairr=residuals(hairind,type="response")
> xtabs(hairr~.,cbind(hair[,1:3],hairr))
, , Sex = Male
       Eye
Hair          Brown        Blue       Hazel       Green
  Black  13.0849617  -7.4851511   2.0040974  -2.5025566
  Brown   2.9101762   1.0485813   3.8256654   0.4284149
  Red    -2.4348863  -2.1522753   1.7434344   3.3825785
  Blond -19.2426840   8.2628316  -4.4025891   1.5294010
, , Sex = Female
       Eye
Hair          Brown        Blue       Hazel       Green
  Black  14.7799032 -11.7378219  -3.9703136  -4.1731191
  Brown   9.8060400 -20.9168246   5.2452805  -2.3473338
  Red     2.0497512  -6.6331977   1.1028494   2.9417458
  Blond -20.9532620  39.6138576  -5.5484244   0.7408692
> mosaicplot(HairEyeColor, shade=T, margin=list(1,2,3))
> hairlm=loglin(HairEyeColor, list(1, 2, 3))
2 iterations: deviation 5.684342e-14
> pchisq(hairlm$pearson, hairlm$df, lower.tail = FALSE)
[1] 5.320872e-23
```

## Extended mosaic plot - with shading

The *extended mosaic plot* shows also the residuals from the log-linear fit, by shading; the fit is done not by `glm()`, but using function `loglin()`, which is handy when we want to see the result of the $\chi^2$ test for independence. We can check that the predictions are the same:

```
> haipar=loglin(HairEyeColor,margin=list(1,2,3),para=T)$para
2 iterations: deviation 5.684342e-14
> as.vector(exp(outer(outer(outer(haipar[[1]],haipar[[2]],"+"),
+ haipar[[3]],"+"),haipar[[4]],"+")))
 [1] 18.915038 50.089824 12.434886 22.242684 18.485151 48.951419 12.152275
 [8] 21.737168  7.995903 21.174335  5.256566  9.402589  5.502557 14.571585
[15]  3.617421  6.470599 21.220097 56.193960 13.950249 24.953262 20.737822
[22] 54.916825 13.633198 24.386142  8.970314 23.754719  5.897151 10.548424
[29]  6.173119 16.347334  4.058254  7.259131
```

# The plot


HairEyeColor
Black
Male
Female
Brown
Male
Female
Red
Male
Female
Blond
Male
Female
Eye
Brown
Blue
Hazel
Green
Hair
>4
2:4
0:2
-2:0
-4:-2
<-4
Standardized Residuals:

## What did we achieve?

We cans see, among other things, that there are more brown-haired, black-eyed individuals and more blond-haired, blue-eyed individuals (especially women) among statistics students than the independence model would suggest. On the other hand, blue-eyed, black- and brown-haired females are underrepresented. Also blondes with brown eyes are rare.

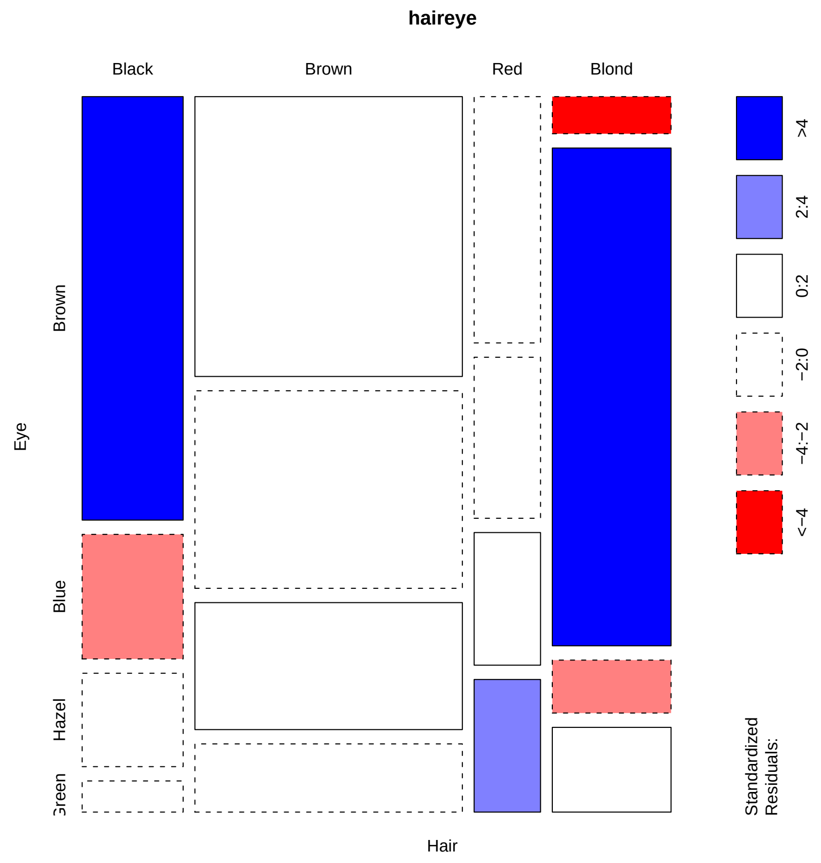
A technical detail: the shading in the mosaic plot works not with "raw" (`type="response"`) residuals, but "Pearson" residuals (`type="pearson"`)

$$\frac{\mathrm{O} - \mathrm{P}}{\sqrt{\mathrm{P}}} \qquad \mathrm{O} \text{ being observed and } \mathrm{P} \text{ predicted frequency}$$

## Check it out

```
> hairpea=residuals(hairind,type="pearson")
> as.numeric(hairpea)
 [1]  3.0086304  0.4111919 -0.6904906 -4.0801117 -1.7409611  0.1498716 -0.6174034  1.7722599
 [9]  0.7087370  0.8313848  0.7604218 -1.4357685 -1.0668458  0.1122306  1.7784773  0.6012418
[17]  3.2084695  1.3081238  0.5487950 -4.1945752 -2.5775431 -2.8225591 -1.7964870  8.0218693
[25] -1.3256260  1.0762019  0.4541456 -1.7083462 -1.6796100 -0.5805657  1.4602779  0.2749788
> as.numeric((hair$Freq-hairpr)/sqrt(hairpr))
 [1]  3.0086304  0.4111919 -0.6904906 -4.0801117 -1.7409611  0.1498716 -0.6174034  1.7722599
 [9]  0.7087370  0.8313848  0.7604218 -1.4357685 -1.0668458  0.1122306  1.7784773  0.6012418
[17]  3.2084695  1.3081238  0.5487950 -4.1945752 -2.5775431 -2.8225591 -1.7964870  8.0218693
[25] -1.3256260  1.0762019  0.4541456 -1.7083462 -1.6796100 -0.5805657  1.4602779  0.2749788
> hairq=as.numeric(cut(hairpea,c(-Inf,-4,-2,0,2,4,Inf)))
> xtabs(hairq~.,cbind(hair[,1:3],hairq))
, , Sex = Male
       Eye
Hair    Brown Blue Hazel Green
  Black     5    3     4     3
  Brown     4    4     4     4
  Red       3    3     4     4
  Blond     1    4     3     4
, , Sex = Female
       Eye
Hair    Brown Blue Hazel Green
  Black     5    2     3     3
  Brown     4    2     4     3
  Red       4    3     4     4
  Blond     1    6     3     4
```

# Aggregated data give similar result



```
> haind=loglin(haireye, list(1, 2))
2 iterations: deviation 0
> pchisq(haind$pearson, haind$df, lower.tail = FALSE)
[1] 2.325287e-25
```

For qualitative data, it is quite easy to get the hypothesis of independence rejected... but how to find out more?

## How to find out more

Log-linear modeling is a way to go; for instance, we can look at the model that sex is independent conditionally on hair and eye colors. What does that mean? Color of eyes and hair are dependent, but there is no dependence between these two and the sex: the chance of encountering a blue-eye blonde man is given by the proportion of blue-eyed blondes (man or women) and the proportion of men about among statistics students

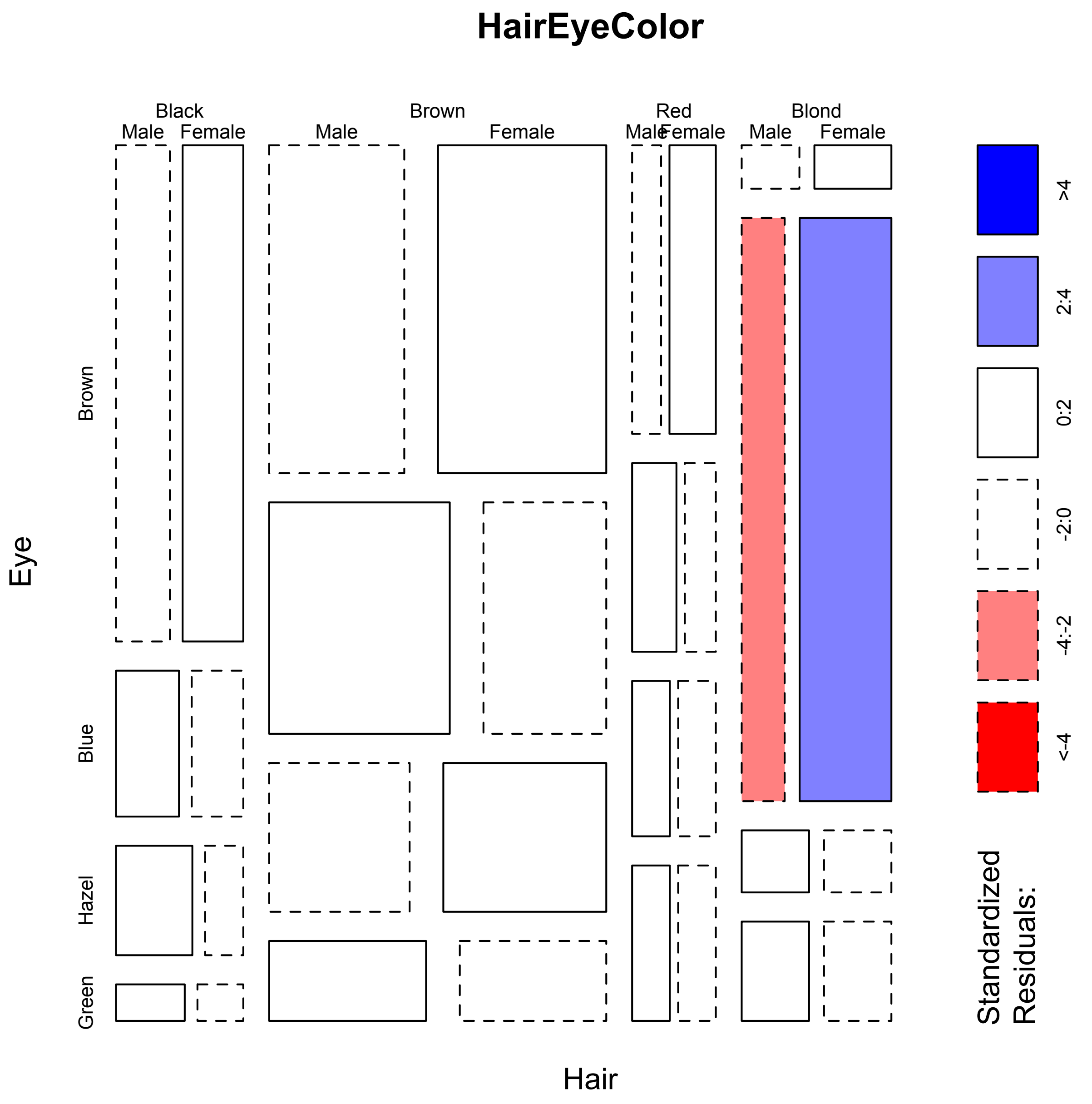
```
> haicind = loglin(HairEyeColor,margin=list(c(1,2),3))
2 iterations: deviation 5.684342e-14
> pchisq(haicind$pearson, haicind$df, lower.tail = FALSE)
[1] 0.1891745
> mosaicplot(HairEyeColor,margin=list(c(1,2),3),shade=T)
```

The plot shows that this fits much better; the most of the discrepancy (pretty much only important from the fitted model) is due to blonde and blue-eyed men and women.

Lack of dependence $\equiv$ lack of (nonzero) interaction

Another way (other than log-linear modeling) to investigate the roots of dependence: correspondence analysis (see later)

## Sex vs. hair-eye independence model



# What is the corresponding glm model?

```
> haircind=glm(formula = Freq ~ Hair*Eye + Sex, family = poisson, data = hair)
> as.numeric(predict(haircind,type="response"))
 [1] 32.047297 56.082770 12.253378  3.298986  9.425676 39.587838  8.011824 44.300676  7.069257
[10] 25.449324  6.597973  4.712838  2.356419 13.667230  6.597973  7.540541 35.952703 62.917230
...
> haicpar=loglin(HairEyeColor,margin=list(c(1,2),3),par=T)$param
> as.vector(exp(outer(outer(outer(haicpar[[1]],haicpar[[2]],"+"),
+ haicpar[[3]],"+"),haicpar[[4]],"+")+rep(as.vector(haicpar[[5]]),2)))
 [1] 32.047297 56.082770 12.253378  3.298986  9.425676 39.587838  8.011824 44.300676  7.069257
[10] 25.449324  6.597973  4.712838  2.356419 13.667230  6.597973  7.540541 35.952703 62.917230
...
> haircr=residuals(haircind,type="pearson")
> haircq=as.numeric(cut(haircr,c(-Inf,-4,-2,0,2,4,Inf)))
> xtabs(haircq~.,cbind(hair[,1:3],haircq))
, , Sex = Male
       Eye
Hair    Brown Blue Hazel Green
  Black     3    4     4     4
  Brown     3    4     3     4
  Red       3    4     4     4
  Blond     3    2     4     4
, , Sex = Female
       Eye
Hair    Brown Blue Hazel Green
  Black     4    3     3     3
  Brown     4    3     4     3
  Red       4    3     3     3
  Blond     4    5     3     3
```

## Dimension Reduction for Quantitative Data: Principal Components

## Something more sophisticated

**Principal components**: aim at finding the linear combinations (in fact, projections) showing the data maximally spread

The measure of spread used here is variance: that is, *principal components look for an expression of the data in the alternative basis of the data space in which the data exhibit maximal variance(s)*

The first definition (equivalent to the current one, by Hotelling) was given by Pearson (1901). The definitions were fine, but had to wait 50 years to become computationally feasible!

Note: *principal*, not principle (principled is a bit different)

Also PC's, if you wish (I don't). Guess what PCA stands for...

Also: maximizing variance is equivalent to maximizing its square root, standard deviation

In more detail:

## The definition of principal components

1. A **first principal (component) direction** is a unit vector $\mathbf{a}_1$, maximizing the (sample) variance of the projection into its direction: maximizing $s^2_{\mathbf{Ya}_1} = \mathbf{a}_1^\top \mathbf{S}_\mathbf{Y} \mathbf{a}_1$ under $\mathbf{a}_1^\top \mathbf{a}_1 = 1 \quad (\mathbf{S}_\mathbf{Y} = \mathrm{var}(\mathbf{Y}))$

Note: clearly, *it is determined uniquely up to a sign*

A **first principal component** is then that projection, $\mathbf{Ya}_1$

2. A **second principal (component) direction** is a unit vector $\mathbf{a}_2$, orthogonal to $\mathbf{a}_1$, maximizing the (sample) variance of the projection into its direction: maximizing $s^2_{\mathbf{Ya}_2} = \mathbf{a}_2^\top \mathbf{S}_\mathbf{Y} \mathbf{a}_2$ under $\mathbf{a}_2^\top \mathbf{a}_2 = 1$ and $\mathbf{a}_1^\top \mathbf{a}_2 = 0$

A **second principal component** is then that projection, $\mathbf{Ya}_2$

3. **A third principal (component) direction** is a unit vector $\mathbf{a}_2$, orthogonal to $\mathbf{a}_1$ and $\mathbf{a}_2$, maximizing the (sample) variance of the projection into its direction: maximizing $s^2_{\mathbf{Ya}_3} = \mathbf{a}_3^\top \mathbf{S}_\mathbf{Y} \mathbf{a}_3$ under $\mathbf{a}_3^\top \mathbf{a}_3 = 1$ and $\mathbf{a}_1^\top \mathbf{a}_3 = 0$, $\mathbf{a}_2^\top \mathbf{a}_3 = 0$,

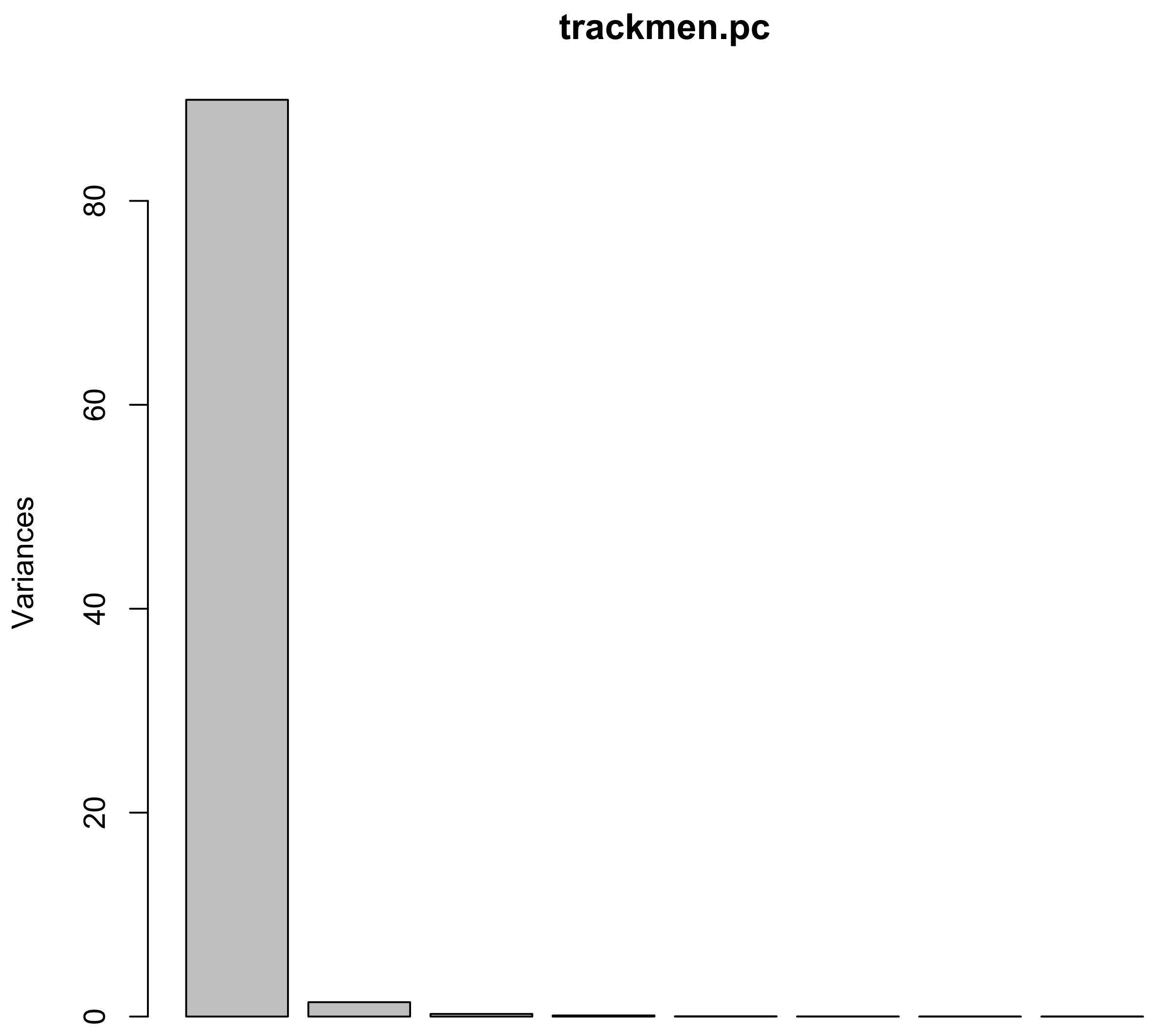
**The third principal component** is then that projection, $\mathbf{Ya}_3$

... and so on, up to... how many? - $p$ principal components

## Let us try it on the male track data

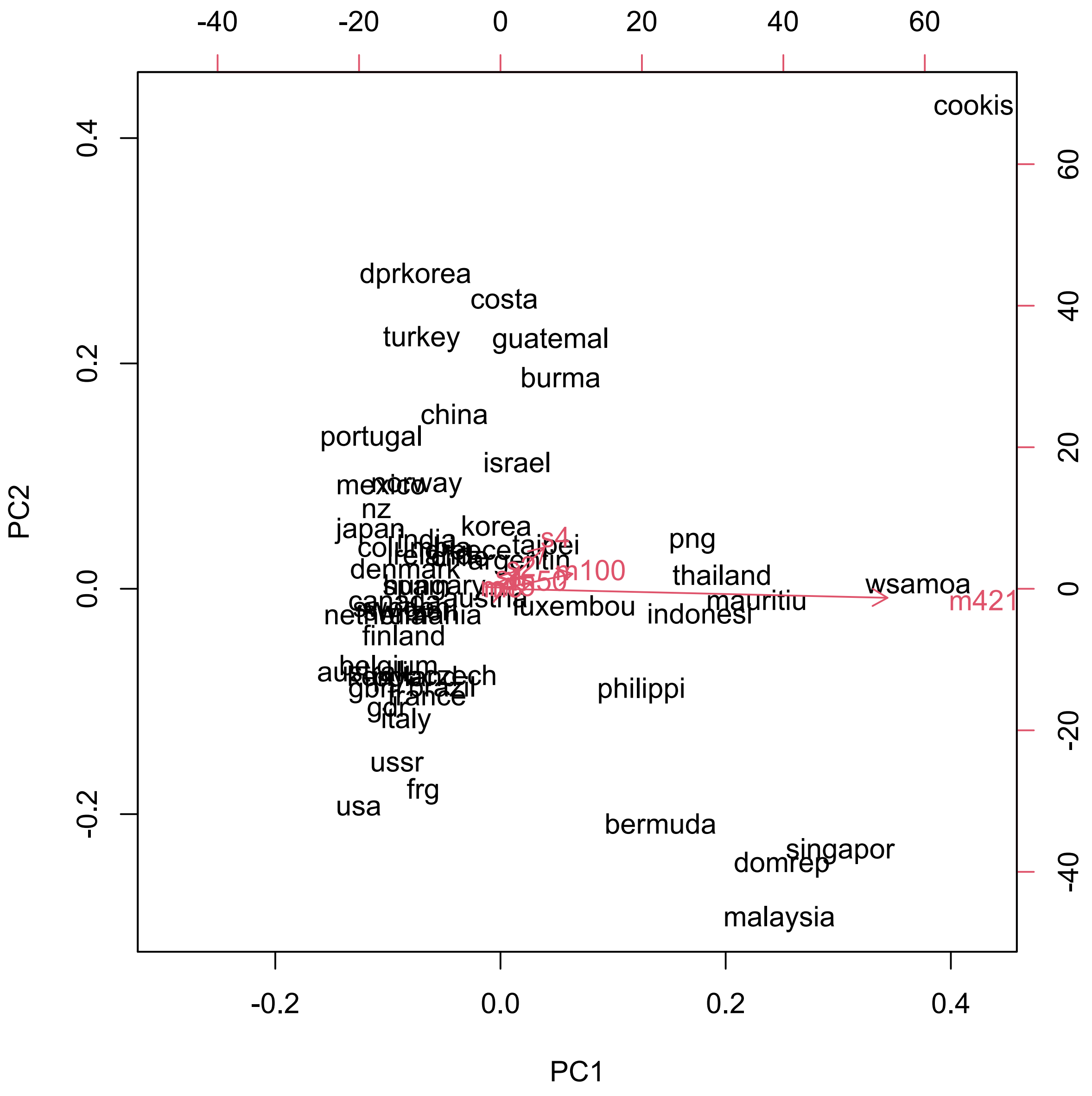
```
> trackmen.pc <- prcomp(trackmen)
> summary(trackmen.pc)
Importance of components:
                          PC1    PC2     PC3     PC4    PC5     PC6
Standard deviation     9.4823 1.1885 0.50975 0.33079 0.1652 0.11284
Proportion of Variance 0.9801 0.0154 0.00283 0.00119 0.0003 0.00014
Cumulative Proportion  0.9801 0.9955 0.99834 0.99953 0.9998 0.99997
> trackmen.pc
Standard deviations (1, .., p=8):
[1] 9.48227935 1.18853948 0.50974911 0.33078740 0.16522894 0.1128397

Rotation (n x k) = (8 x 8):
             PC1         PC2          PC3          PC4          PC5
s1   0.019865407  0.21068958 -0.029041979  0.358784470  0.190181784
s2   0.041554499  0.35892579 -0.018390126  0.833534544 -0.048582165
s4   0.110631838  0.82786251 -0.377669011 -0.396041212 -0.012020033
m8   0.005487699  0.02317490  0.005341591  0.009568087 -0.011107487
m15  0.014386822  0.04465255  0.050004337  0.015981502 -0.043222520
m50  0.079308444  0.12996134  0.336448522 -0.018873808 -0.909186992
m100 0.181098994  0.29885393  0.848722695 -0.134662690  0.364239482
m421 0.972787446 -0.18080736 -0.141872114  0.028425488  0.006575083
```

## The screeplot



```
> plot(trackmen.pc)
```

# And possibly also this: the biplot



```
> biplot(trackmen.pc)
```

## Two functions in R: `prcomp()` and `princomp()`

They return quite similar results, but there are also differences

We have just seen `prcomp()`; now `princomp()`

```
> trackmen.pc2 <- princomp(trackmen)
> trackmen.pc2$loadings

Loadings:
     Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8
s1           0.211         0.359  0.190  0.887              
s2           0.359         0.834        -0.410              
s4    0.111  0.828  0.378 -0.396                            
m8                                              0.261 -0.965
m15                                             0.959  0.262
m50          0.130 -0.336        -0.909  0.184              
m100  0.181  0.299 -0.849 -0.135  0.364                     
m421  0.973 -0.181  0.142                                   
...
```

“Loadings” = principal component directions

## Principal components: computation I[3]

Principal components can be obtained via eigenvector-eigenvalue decomposition applied to the variance-covariance matrix (or, in scaled case, correlation matrix). Principal component directions are given by the eigenvectors; eigenvalues give the variance of the principal component. This is what R function `princomp()` does

Note that this method of computation does not require the knowledge of the original dataset; principal components can be computed merely from the variance-covariance or correlation matrix - and it is indeed possible in `princomp()` to do it this way

If `princomp()` uses the original data, it computes the variance-covariance matrix first, and then applies the eigenvector-eigenvalue decomposition to this matrix

[3]See Problem 8 for the technical details

## Principal components: computation II[4]

Another way is to avoid the computation of the variance-covariance or correlation matrix and compute principal components directly by applying singular value decomposition (SVD) to the *centered* data matrix, the data matrix with column averages equal to zero

$$\tilde{\mathbf{Y}} = \mathbf{Y} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top\mathbf{Y}$$

This is how it is done by the R function `prcomp()`

It is also reason why many applied sources don't refer explicitly to the method as "principal components" but rather speak about "applying SVD to the data matrix"

Note, however (important!): if the data matrix *not* centered, principal components *won't* result

[4]See Problem 20 for the technical details

## Principal components: issues

Remember: principal components *do change* after changing the scale of variables (not after shift and rotation/orthogonal transformation, however) - and do not change in a way we'd like to see (= they are not *equivariant*)

Note: computing principal components out of the original data (that is: out of the variance-covariance matrix) often results merely in emphasizing the "size factor" - as we have seen on the example above

Thus, to mitigate the just mentioned drawback, it may be preferable to compute principal components from the *scaled* data: we subtract the means of the columns of $\mathbf{Y}$ from the corresponding columns (obtaining the centered data matrix $\tilde{\mathbf{Y}}$), and then we also divide those columns by their corresponding standard deviations

*Computing principal components out of the variance-covariance matrix of the standardized data is equivalent to computing them directly out of the correlation matrix*

Principal components are rather sensitive to outliers (that is, they are in general not robust)

## Principal components: pseudo-issues

Of course, everything above can become controversial from the purely mathematical point: if two eigenvalues coincide, then any linear combination of the corresponding eigenvectors is again an eigenvector for this particular eigenvalues, and we are running into mathematical difficulties: principal directions are defined ambiguously (and not just up to the sign)

However, this kind of difficulty may be quite ignored from the practical point of view: for the real-world data, the data that are *in general position*, the exact equality of eigenvalues never occurs

On the other hand though, some underlying spherical structure may still wreak havoc the analysis: for instance, if the data are sampled from a perfect sphere, then eigenvalues will be different and principal directions therefore unique, but a small perturbation of the input may change the results dramatically. One thus has to be on alert regarding the effects of this kind: the ambiguity in perfect mathematical setting typically translates to an unstability, in the setting that is numerically approximate

## Principal components: pseudo-issues continued

In a similar fashion, it is very rare to encounter an eigenvalue *exactly* equal to 0. Such an eigenvalue in mathematical setting means *degeneracy*, data being concentrated on an affine subspace, one variable typically being a linear or affine combination of others. This can create problems in some other analyses, but in the eigenvalue context, it just demonstrates itself by an eigenvalue that is *very small*

This can be illustrated on a small experiment in R with the male track data. Adding a variable which is the sum of all other variables to the dataset certainly creates degeneracy; the variance-covariance matrix will be singular, not invertible

```
> solve(var(cbind(trackmen,apply(trackmen,1,sum))))
Error in solve.default(var(cbind(trackmen, apply(trackmen, 1, sum)))) :
  system is computationally singular: reciprocal condition number = 4.53665e-18
```

Nonetheless, the eigenvalue decomposition, and subsequently the algorithms for principal components will proceed through:

## Indeed

```
> eigen(var(cbind(trackmen,apply(trackmen,1,sum))))
eigen() decomposition
$values
[1] 2.754561e+02 2.783394e+00 2.784531e-01 1.170712e-01 2.815015e-02 1.313185e-02 2.607990e-03 4.687135e-04 6.143534e
...
> princomp(cbind(trackmen,apply(trackmen,1,sum)))
Call:
princomp(x = cbind(trackmen, apply(trackmen, 1, sum)))

Standard deviations:
      Comp.1       Comp.2       Comp.3       Comp.4       Comp.5       Comp.6       Comp.7       Comp.8       Comp.9
1.644530e+01 1.653114e+00 5.228674e-01 3.390319e-01 1.662478e-01 1.135477e-01 5.060210e-02 2.145207e-02 7.416130e-08

 9  variables and  55 observations.

> prcomp(cbind(trackmen,apply(trackmen,1,sum)))
Standard deviations (1, .., p=9):
[1] 1.659687e+01 1.668351e+00 5.276866e-01 3.421567e-01 1.677801e-01 1.145943e-01 5.106849e-02 2.164979e-02 4.597450e
...
```

Thus, when doing mathematics of principal components and related concepts, it is pretty much safe ("without loss of generality") to assume the data in general position, translating to all eigenvalues being positive and distinct

## Principal components: uses

Overall, one can use of principal components to condense the information in the data - to reduce the dimension: we retain only certain number of the (initial) principal components

This number can be specified in advance (for instance, in information transmission), or one can be guided by the size of variance: the ratio of a variance to the sum of all variances gives "proportion of variance explained". A closely-related indicator is the "cumulative proportion of variance explained"; standard deviations are used too. A graphical tool to facilitate this, showing maximized variance (or standard deviation) for each principal component, is called **screeplot**

Another potential use: finding interesting projections of the data

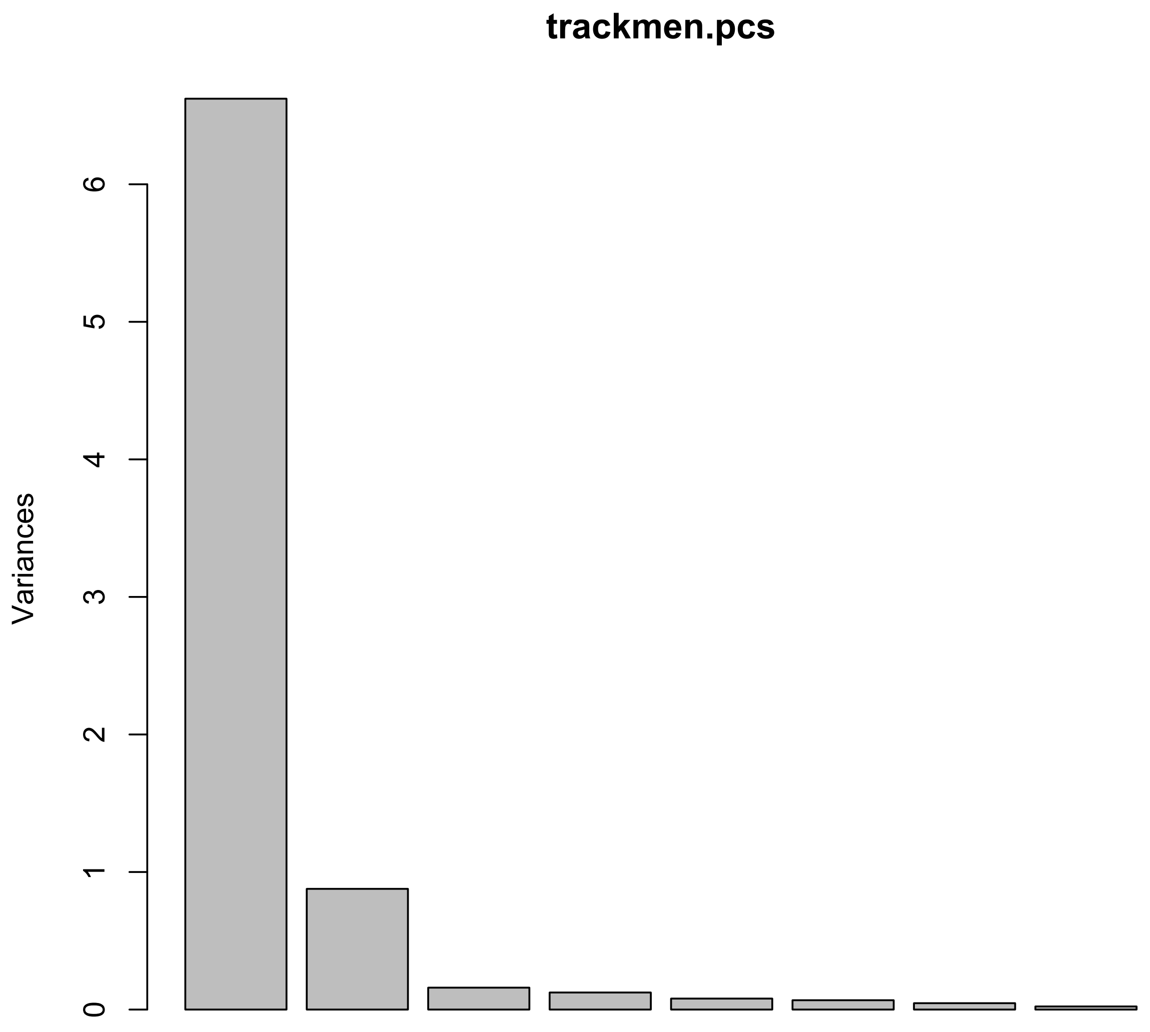
## Like this

```
> trackmen.pcs = prcomp(trackmen,scale=TRUE)
> summary(trackmen.pcs)
Importance of components:
                          PC1    PC2     PC3     PC4     PC5    PC6
Standard deviation     2.5734 0.9368 0.39915 0.35221 0.28263 0.2607
Proportion of Variance 0.8278 0.1097 0.01992 0.01551 0.00999 0.0085
Cumulative Proportion  0.8278 0.9375 0.95739 0.97289 0.98288 0.9914
```

Same as

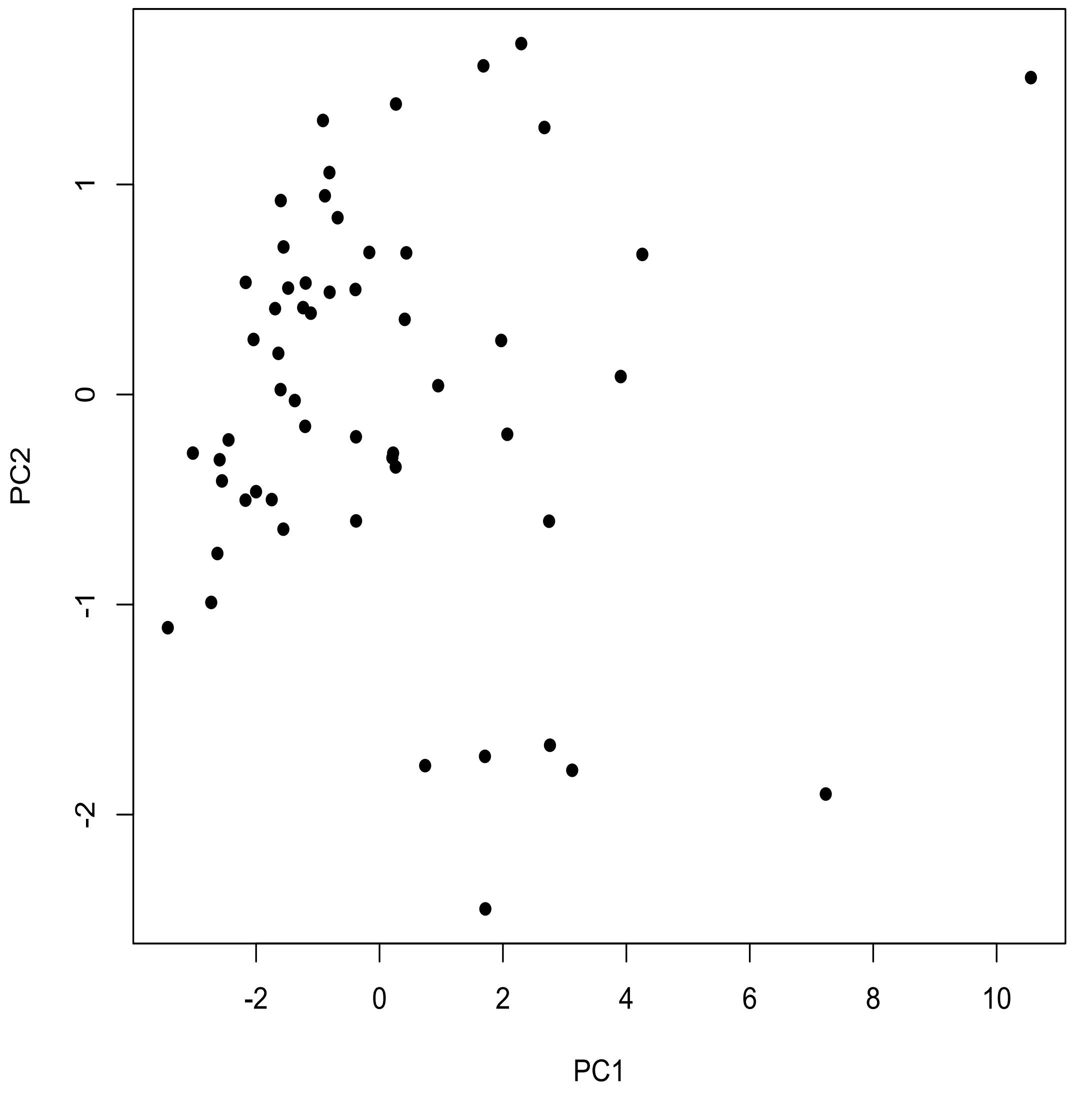
```
> trackmen.pcs = prcomp(scale(trackmen))
> summary(trackmen.pcs)
Importance of components:
                          PC1    PC2     PC3     PC4     PC5    PC6
Standard deviation     2.5734 0.9368 0.39915 0.35221 0.28263 0.2607
Proportion of Variance 0.8278 0.1097 0.01992 0.01551 0.00999 0.0085
Cumulative Proportion  0.8278 0.9375 0.95739 0.97289 0.98288 0.9914
```

## So, another screeplot



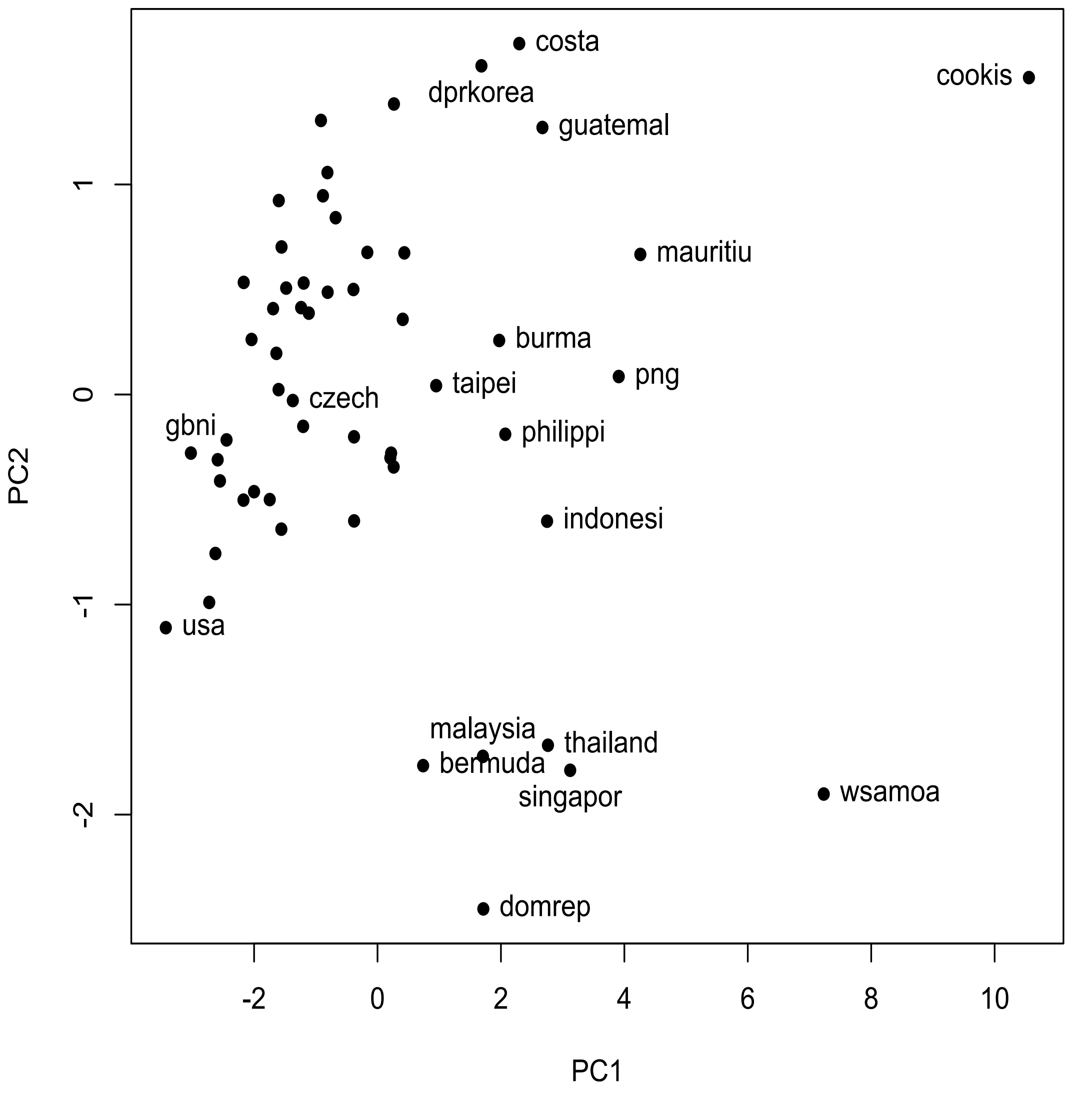
```
> plot(trackmen.pcs)
```

## We may plot the first two components now



```
> plot(predict(trackmen.pcs),pch=16)
```

# With some additional information perhaps

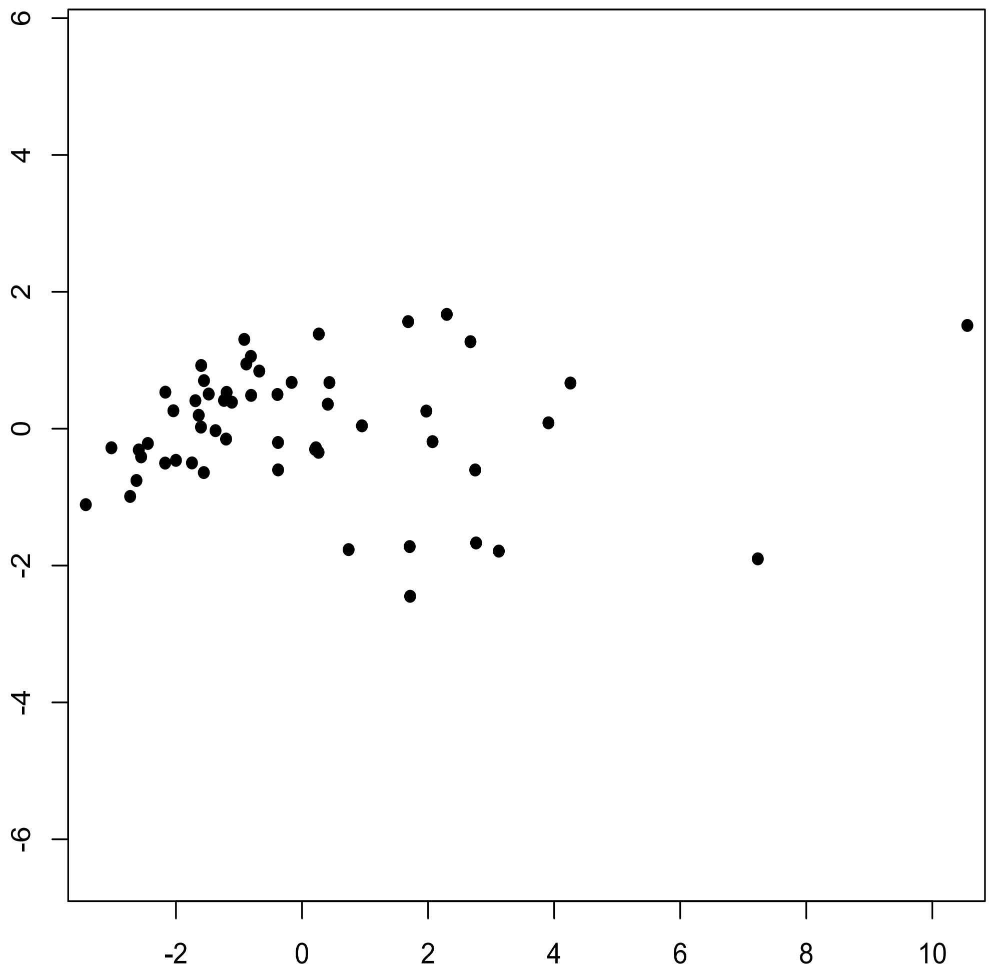


```
> plot(predict(trackmen.pcs),pch=16)
```

## And look more into them

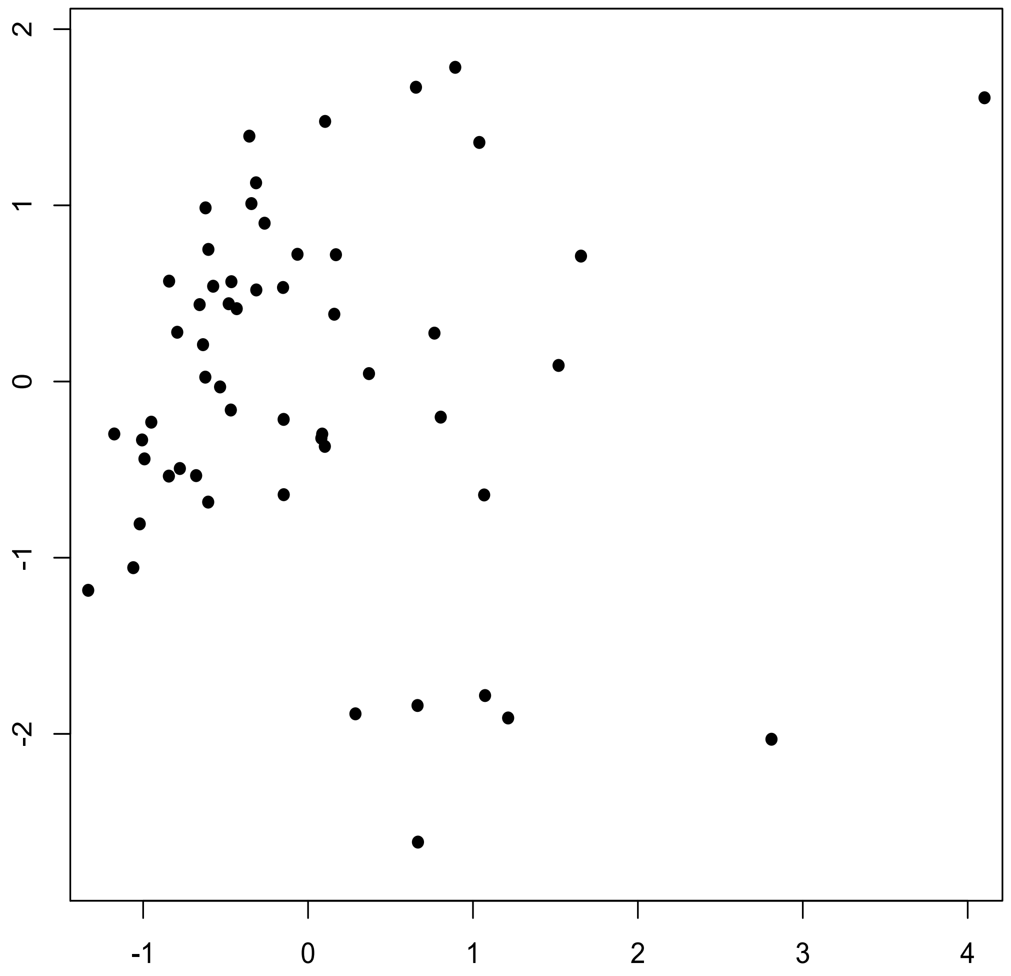
```
> trackmen.pcs
Standard deviations (1, .., p=8):
[1] 2.5733531 0.9368128 0.3991505 0.3522065 0.2826310 0.2607013 0.21
[8] 0.1503333

Rotation (n x k) = (8 x 8):
           PC1         PC2        PC3         PC4        PC5
s1   0.3175565  0.56687750  0.3322620 -0.12762827  0.2625555 -0.5937
s2   0.3369792  0.46162589  0.3606567  0.25911576 -0.1539571  0.6561
s4   0.3556454  0.24827331 -0.5604674 -0.65234077 -0.2183229  0.1566
m8   0.3686841  0.01242993 -0.5324823  0.47999895  0.5400528 -0.0146
m15  0.3728099 -0.13979665 -0.1534427  0.40451039 -0.4877151 -0.1578
m50  0.3643741 -0.31203045  0.1897643 -0.02958755 -0.2539792 -0.1412
m100 0.3667726 -0.30685985  0.1817517 -0.08006862 -0.1331764 -0.2190
m421 0.3419261 -0.43896267  0.2632087 -0.29951213  0.4979283  0.3152
```

## Arguably, equiscaled plot is more appropriate



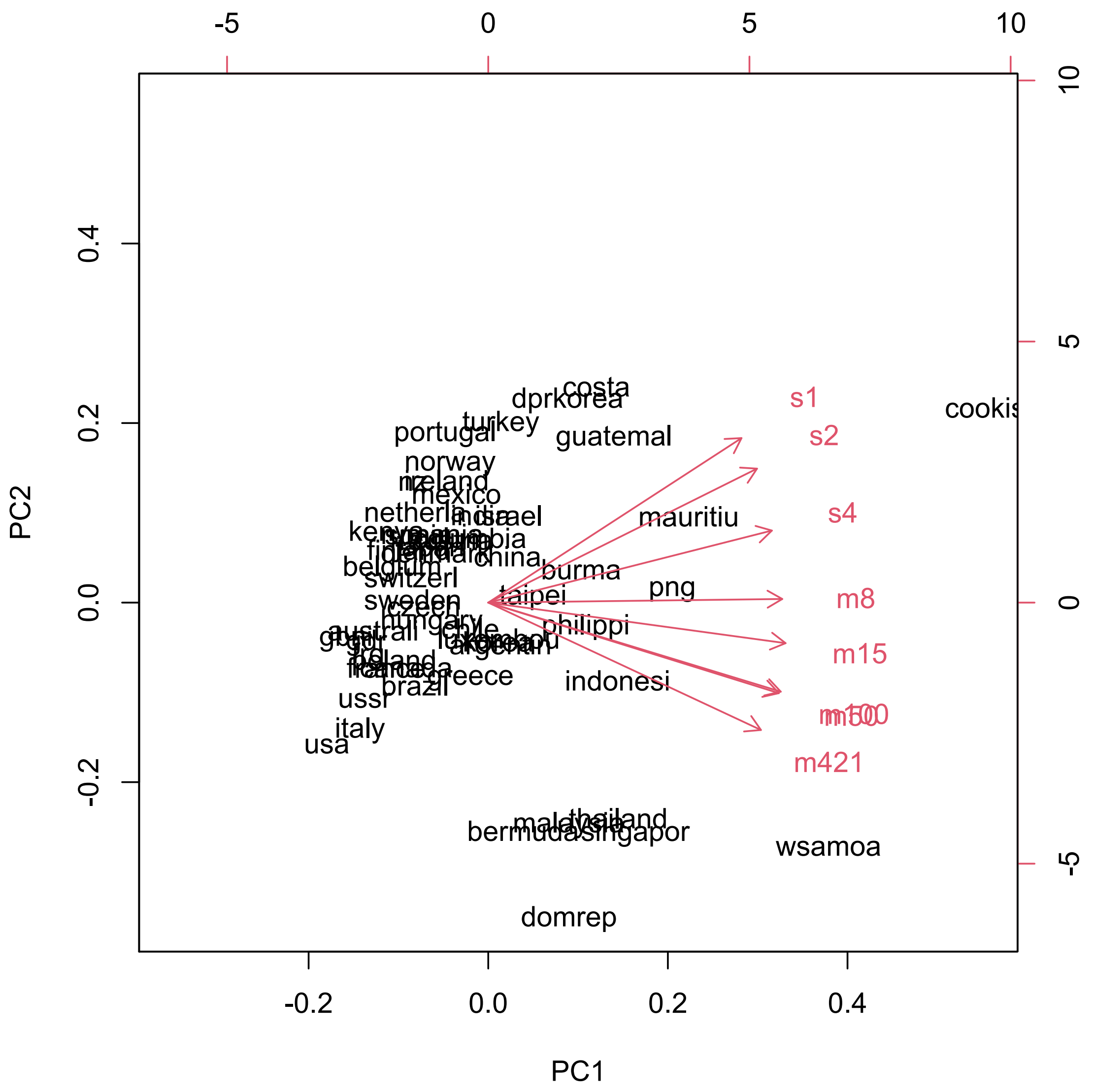
```
> library(MASS)
> eqscplot(predict(trackmen.pcs)[,1:2],pch=16)
```

## A potential half-way: orthonormal components



```
> library(MASS)
> eqscplot(sweep(predict(trackmen.pcs)[,1:2],2,
+ trackmen.pcs$sdev[1:2],"/"),pch=16)}
```

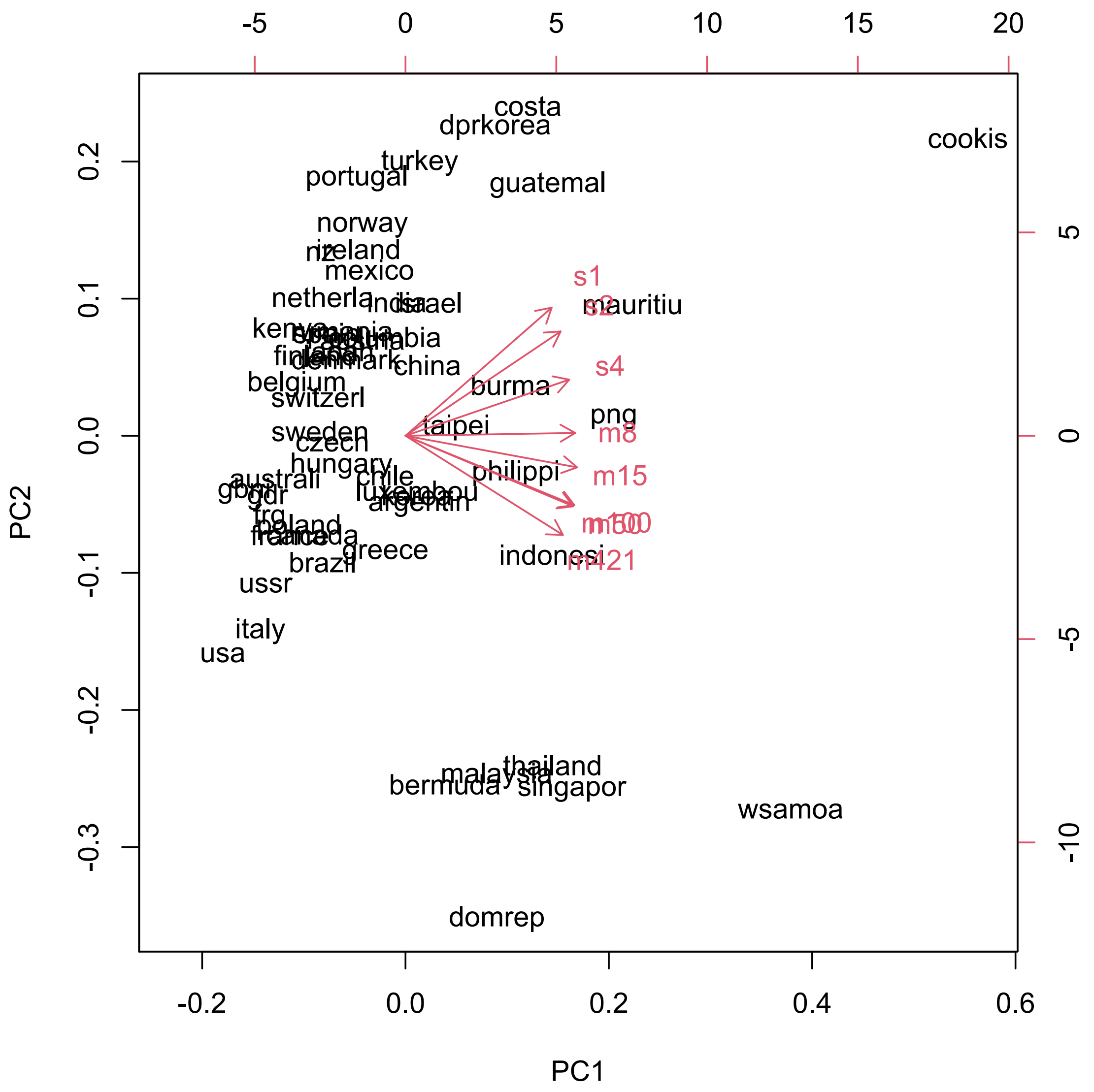
## But usually we rather use a biplot



```
> biplot(trackmen.pcs)
```

originally equiscaled...

## But usually we rather use a biplot



```
> biplot(trackmen.pcs,xlim=c(-0.23,0.57))
```

…then somewhat rescaled for aesthetic reasons

## Biplots of principal components

Biplots handle very well the situation when the first two principal components are plotted (which is often the case)

What is there in “bi”? First, we plot the principal components, the 2-element vectors formed by the corresponding elements of $\mathbf{Ya}_1$ and $\mathbf{Ya}_2$. They correspond to the rows of the original data matrix, which implies their possible labels

The we also plot the 2-element vectors formed by corresponding elements of $\mathbf{a}_1$ and $\mathbf{a}_2$ - to distinguish them from the previously plotted points, we use arrows. These arrows correspond to the columns of the original data matrix, the variables

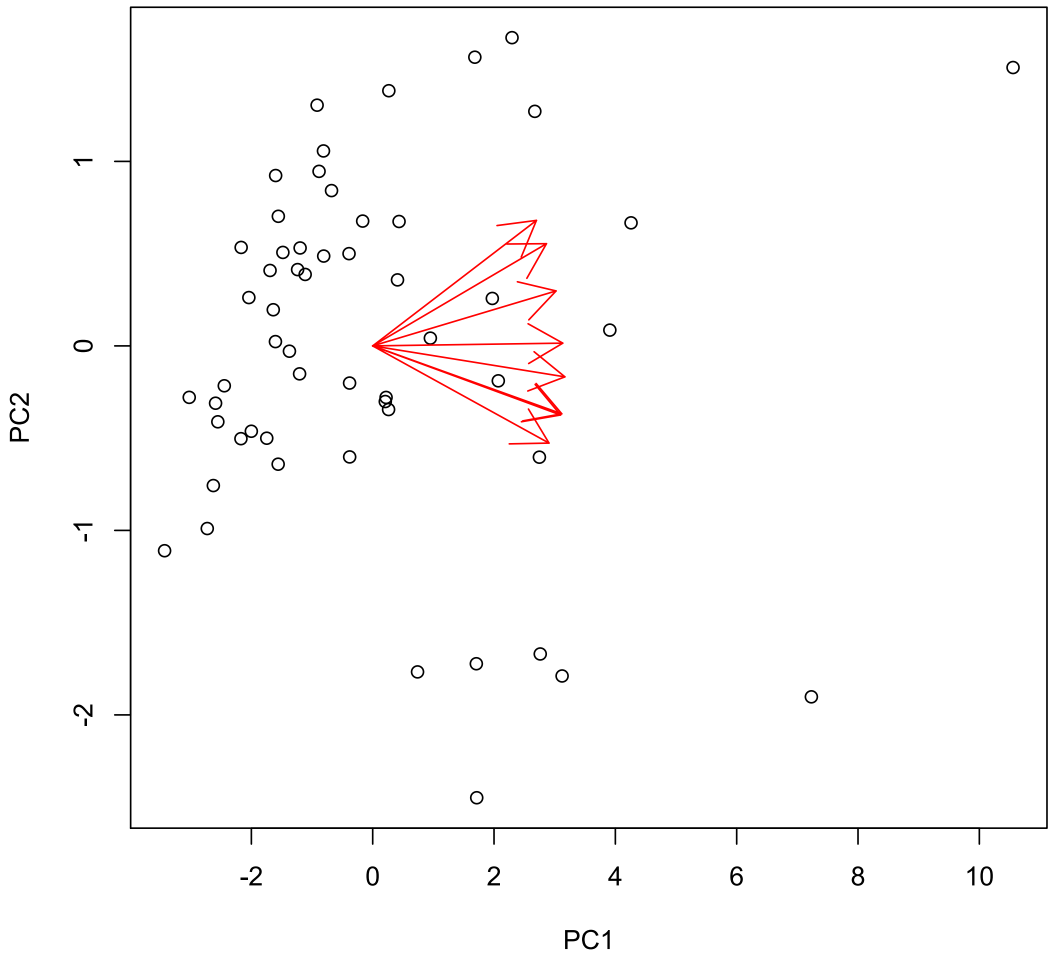
Some rescaling has usually to take place, so that we can see how the first two principal components relate to the original variables

## Check it out: first, some look into the R output

```
> attributes(trackmen.pcs)
$names
[1] "sdev"     "rotation" "center"   "scale"    "x"

$class
[1] "prcomp"

> trackmen.pcs$rotation
           PC1         PC2        PC3         PC4        PC5        PC6
s1   0.3175565  0.56687750  0.3322620 -0.12762827  0.2625555 -0.5937042
s2   0.3369792  0.46162589  0.3606567  0.25911576 -0.1539571  0.6561367
s4   0.3556454  0.24827331 -0.5604674 -0.65234077 -0.2183229  0.1566252
m8   0.3686841  0.01242993 -0.5324823  0.47999895  0.5400528 -0.0146918
m15  0.3728099 -0.13979665 -0.1534427  0.40451039 -0.4877151 -0.1578430
m50  0.3643741 -0.31203045  0.1897643 -0.02958755 -0.2539792 -0.1412987
m100 0.3667726 -0.30685985  0.1817517 -0.08006862 -0.1331764 -0.2190168
m421 0.3419261 -0.43896267  0.2632087 -0.29951213  0.4979283  0.3152849

> plot(predict(trackmen.pcs))
> arrows(0,0,trackmen.pcs$rotation[,1],trackmen.pcs$rotation[,2])
```

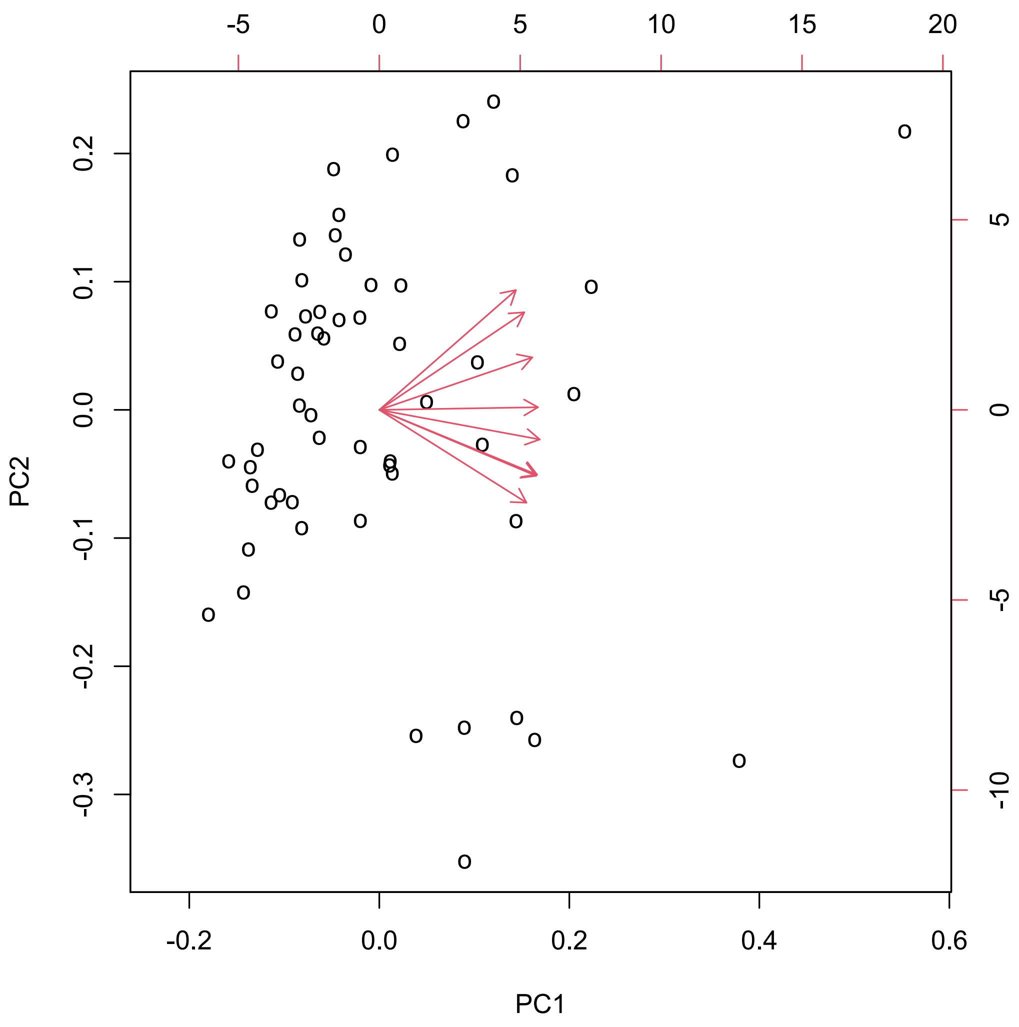
It is not really it yet - but after some rescaling...

## Pretty close, right?



```
> plot(predict(trackmen.pcs))
> arrows(0,0,8.5*trackmen.pcs$rotation[,1],
+ 1.2*trackmen.pcs$rotation[,2],col="red")
```

## Compare to the (somewhat edited) R own



## Principal components: summary

Principal components are projections with maximal spread, the projections that maximize the variance in the given direction - the more precise sense is given above. This may reveal interesting projection(s) (although arguably, there may be also more interesting ones)

The original definition of Pearson (1901) used line fitting via least squares minimizing *orthogonal* distances: the first principal component would be the projection of the datapoints in the direction of the line giving the best orthogonal regression fit to the data points (the sum of squared geometric (orthogonal) distances of points to the line is minimal). In other words, first principal component consists of projections of datapoints giving best orthogonal fit of the data.

The best fit of this type should pass through the mean; hence it is reasonable to consider centered datapoints $\mathbf{y}_i - \bar{\mathbf{y}}$.

Subsequent principal components would be defined by analogous fitting in the directions orthogonal to those of previous ones

## Principal components: stochastic underpinning (population principal components)?

As defined above, principal components computed out of the data as indicated above may appear as a purely data-analytic method, a method requiring no stochastic underpinning.

To a large extent it is so; however, if we assume that the data are sampled from a certain distribution, then it is possible to define principal directions in an analogous way out of the variance-covariance or correlation matrix of this distribution

That is, $\mathrm{var}(\mathbf{Ya})$ is replaced by $\mathrm{Var}(\mathbf{a}^\top\mathbf{y})$, where $\mathbf{y}$ is a random vector. Singular value decomposition strategy is not available here, but we can still do the eigenvalue decomposition of the stochastic variance-covariance matrix $\mathrm{Var}(\mathbf{y})$, to obtain the pertinent solutions.

If the lines of the data matrix $\mathbf{Y}$ are then considered to be independently sampled from the distribution of $\mathbf{y}$, then the principal components computed out of $\mathbf{Y}$ can be considered "sample principal components" - estimates of the previous stochastic ("population") principal components

## Ramifications

It is important to understand the definition of principal components, because there are quite a few other procedures that elaborate on their idea. Some of them are:

- common principal components

- principal curves and surfaces
  (Pearson: the first principal component is the orthogonal regression fit to the data)

- self-organized mappings (Kohonen)

- independent component analysis (ICA)
  (note: principal components are uncorrelated)

- partial least squares (PLS)
  ("principal components of the regressors with view on a response")

- projection pursuit

## More about dimension reduction in regression

We have a linear regression (univariate response)

$$\mathbf{y} \sim \mathbf{X}\boldsymbol{\beta} \qquad \text{(or, if you wish, } \hat{\mathbf{y}} = \mathbf{X}\boldsymbol{\beta}\text{)}$$

This is nice for general theory, but in practice it is useful to separate intercept out of **X** (we will abuse the notation and denote the new matrix also by **X**)

$$\mathbf{y} \sim \mathbf{1}\alpha + \mathbf{X}\boldsymbol{\beta}$$

We may assume **y**, **X** centered (otherwise the parameters transform how?) That is, the average of the elements of **y** is 0, as well as the averages of the all columns of **X**. (Then, if we are doing the least-squares fit, the coefficient $\alpha$ would be 0)

We can also assume the columns of **X** scaled (otherwise we may again transform the fitted parameters in the appropriate way)

But **X** may have way too many columns; we would like to use fewer of those...

## Principal component regression

We compute principal components **Z** out of **X** and retain only first $m$ of those, $\mathbf{Z}_m$

$$\mathbf{y} \sim \mathbf{1}\alpha + \mathbf{Z}_m\gamma$$

Now, we know that the columns of $\mathbf{Z}_m$ are orthogonal - when the fitting criterion is least squares, we obtain the solution by solving $m$ univariate regressions

$$\mathbf{y} \;\; \mathbf{z}_j\gamma_j$$

Of course, when $M = p$, that is, we use all principal components, we get only the solution of the original regression, nothing new

The just described strategy was and is used, with possible advantages and drawbacks. Rather than going into those, let us mention one fact: principal components do not care about **y**; for every **y** they are the same. This leads to the method of partial least squares

## Partial least squares (PLS)

Partial least squares are "principal components for regression". This is best seen from the following analogy

Principal component *directions* $\mathbf{a}_m$ are found via maximizing

$$\mathrm{var}(\mathbf{X}\mathbf{a}_\mathbf{m}) \quad \text{subject to } \|\mathbf{a}_m\| = 1 \quad \text{and}$$
$$\mathbf{a}_m^\top \mathbf{a}_k = 0 \text{ for all } k = 1, 2, \ldots, m-1$$

The last condition can be rewritten so that the definition is

$$\mathrm{var}(\mathbf{X}\mathbf{a}_\mathbf{m}) \quad \text{subject to } \|\mathbf{a}_m\| = 1 \quad \text{and}$$
$$\mathbf{a}_m^\top \mathbf{S}_\mathbf{X} \mathbf{a}_k = 0 \text{ for all } k = 1, 2, \ldots, m-1$$

where $\mathbf{S}_\mathbf{X} = \mathrm{var}(\mathbf{X})$ is the variance-covariance matrix of $\mathbf{X}$

Now, **partial least squares directions** $\mathbf{b}_m$ are found via maximizing

$$(\mathrm{cor}(\mathbf{y}, \mathbf{X}\mathbf{b}_\mathbf{m}))^2 \,\mathrm{var}(\mathbf{X}\mathbf{b}_\mathbf{m}) \quad \text{subject to } \|\mathbf{b}_m\| = 1 \quad \text{and}$$
$$\mathbf{b}_m^\top \mathbf{S}_\mathbf{X} \mathbf{b}_k = 0 \text{ for all } k = 1, 2, \ldots, m-1$$

In practical applications, only some small number of directions is used, like in principal components. However, the algorithm for partial least squares is sequential

## Projection pursuit

Named by Friedman and Tukey (Kruskal considered earlier under different name). Evolved from “Grand Tour” - a technique that “traveled all” directions to find those yielding interesting projections.

The objective is to find the directions giving most interesting projections of the data. Deemed to be those maximizing some *projection index*.

While principal components are also a kind of projection pursuit, this kind of projection is not considered that interesting here. Hence the datapoints are first scaled so that their covariance matrix is identity: *sphering*. This is equivalent to transforming to principal components (the original ones, computed from the covariance matrix, after subtracting the mean) and then scaling to unit variance (dividing by square root of eigenvalues).

The projections of interest are one- and two-dimensional. There are several indices, mostly quantifying a degree of deviation from normality as a measure of interestingness. Thus, the computation of the index *often involves the estimation of the density of projected points*.

## Examples of projection indices

Indices assessing "non-uniformity" (uniform = uninteresting)

entropy $\int f \log f$

Indices assessing "non-normality" (normal = uninteresting)

skewness and curtosis

variations of $L^2$ deviation from normality

$$\int (f-\varphi)^2/2\varphi \quad \int (f-\varphi)^2 \quad \int (f-\varphi)^2\varphi$$

Indices finding "holes" in the data

Indices finding clusters in the data

(the above rather superimpose them)

## Technical details

Computing involves some kind of numerical minimization.

Once an interesting projection is find, there is a need to perform a *structure removal* before finding another one. The structural removal can be accomplished, for instance, by transforming to normality in the selected projection

Limitations:

Still somewhat in development (that is, was 20 years ago)

The opinions on various indices differ

Computationally demanding, needs sophisticated optimization methods (many local maxima)

Practical impact?

## Reducing Dimension of Hidden causes: Factor Analysis

## Preamble

Attempts of interpreting principal components may lead to the entities with their own lives: factors

It is something like retaining $k$ principal components and interpreting those -

- but beware: the process will be different, and the results - unlike principal components! - will be equivariant with respect to rescaling

The name of the technique (or bundle of techniques) is **factor analysis**, and it can be approached from several viewpoints - one of them is: a sort of reading from the variance-covariance matrix

## Motivation: principal components

Recall: we have the data matrix $\mathbf{Y}$, and we decompose its variance-covariance matrix $\mathbf{S_Y} = \mathbf{U\Lambda U}^\top$.

The principal component directions are columns of $\mathbf{U}$:

$\begin{pmatrix}\mathbf{a}_1 & \mathbf{a}_2 & \dots & \mathbf{a}_p\end{pmatrix} = \begin{pmatrix}\mathbf{u}_1 & \mathbf{u}_2 & \dots & \mathbf{u}_p\end{pmatrix} = \mathbf{U}$

Principal components are the projections into those directions:

$\mathbf{C} = \begin{pmatrix}\mathbf{c}_1 & \mathbf{c}_2 & \dots & \mathbf{c}_p\end{pmatrix} = \begin{pmatrix}\mathbf{Yu}_1 & \mathbf{Yu}_2 & \dots \mathbf{Yu}_p\end{pmatrix} = \mathbf{YU}.$

That is, principal components are expressed as linear combinations of the original variables - the columns of $Y$:

$$\begin{pmatrix} c_{11} & \dots & c_{1p} \\ c_{21} & \dots & c_{2p} \\ \dots & \dots & \dots \\ c_{n1} & \dots & c_{np} \end{pmatrix} = \begin{pmatrix} y_{11} & \dots & y_{1p} \\ y_{21} & \dots & y_{2p} \\ \dots & \dots & \dots \\ y_{n1} & \dots & y_{np} \end{pmatrix} \begin{pmatrix} u_{11} & \dots & u_{1p} \\ u_{21} & \dots & u_{2p} \\ \dots & \dots & \dots \\ u_{p1} & \dots & u_{pp} \end{pmatrix}$$

This is all standard, nothing new here.

## Principal components inverted

We may invert the relationship $\mathbf{YU} = \mathbf{C}$: express the original variables as linear combinations of principal components. It should not be a problem, because $\mathbf{U}^{-1} = \mathbf{U}^\top$.

$$\mathbf{Y} = \mathbf{C}\mathbf{U}^\top = \begin{pmatrix} \mathbf{c}_1 & \dots & \mathbf{c}_p \end{pmatrix} \begin{pmatrix} \mathbf{u}_1^\top \\ \dots \\ \mathbf{u}_p^\top \end{pmatrix} = \mathbf{c}_1\mathbf{u}_1^\top + \cdots + \mathbf{c}_p\mathbf{u}_p^\top$$

However, we usually want only few first principal components (reduction of the information)

$$\begin{aligned} \mathbf{Y} &= \mathbf{c}_1\mathbf{u}_1^\top + \cdots + \mathbf{c}_m\mathbf{u}_m^\top + \mathbf{c}_{m+1}\mathbf{u}_{m+1}^\top + \cdots + \mathbf{c}_p\mathbf{u}_p^\top \\ &= \mathbf{c}_1\mathbf{u}_1^\top + \cdots + \mathbf{c}_m\mathbf{u}_m^\top + \mathbf{E} = \mathbf{C}_m\mathbf{U}_m^\top + \mathbf{E} \end{aligned}$$

This sets the motivation of the factor analysis model. Before proceeding to it, however, we have to address a technicality: centering

## Centering

Return to the equation

$$\begin{pmatrix} c_{11} & \dots & c_{1p} \\ c_{21} & \dots & c_{2p} \\ \dots & \dots & \dots \\ c_{n1} & \dots & c_{np} \end{pmatrix} = \begin{pmatrix} y_{11} & \dots & y_{1p} \\ y_{21} & \dots & y_{2p} \\ \dots & \dots & \dots \\ y_{n1} & \dots & y_{np} \end{pmatrix} \begin{pmatrix} u_{11} & \dots & u_{1p} \\ u_{21} & \dots & u_{2p} \\ \dots & \dots & \dots \\ u_{p1} & \dots & u_{pp} \end{pmatrix}$$

Taking means columnwise shows that

$\begin{pmatrix} \bar{c}_1 & \dots & \bar{c}_p \end{pmatrix} = \begin{pmatrix} \bar{y}_1 & \dots & \bar{y}_p \end{pmatrix} \mathbf{U}$, that is, $\bar{\mathbf{c}}^\top = \bar{\mathbf{y}}^\top \mathbf{U}$.

In plain language: means projects onto means. Hence, if we take $\mathbf{Y}$ centered - instead of $\mathbf{Y}$ we consider $\tilde{\mathbf{Y}} = \mathbf{Y} - \mathbf{1}\bar{\mathbf{y}}^\top$, obtained by subtracting the column means from $\mathbf{Y}$, then the column means of $\mathbf{C}$ are zeros - as are those of $\tilde{\mathbf{Y}}$ (and everything is all right).

Instead of

$$\mathbf{Y} = \mathbf{C}_m \mathbf{U}_m^\top + \mathbf{D}$$

we may assume that $\mathbf{C}_m$ is centered and write instead

$$\tilde{\mathbf{Y}} = \mathbf{C}_m \mathbf{U}_m^\top + \mathbf{D}$$

## Factor model - attempting a purely sample form

All the above could lead to a hypothetic factor model

$$\tilde{\mathbf{Y}} = \mathbf{C}_m \mathbf{U}_m^\top + \mathbf{D} \qquad \text{where} \qquad \tilde{\mathbf{Y}} = \mathbf{Y} - \mathbf{1}\bar{\mathbf{y}}^\top,$$

and $\mathbf{1}$ is the $n \times 1$ column of ones; we could call the $n \times m$ matrix $\mathbf{C}_m$ the matrix of (common) factors (columns of $\mathbf{C}_m$), and $\mathbf{D}$ would be the $n \times p$ matrix of specific factors.

We may also write

$$\mathbf{Y} - \mathbf{1}\bar{\mathbf{y}}^\top = \mathbf{C}_m \mathbf{U}_m^\top + \mathbf{D}$$

and it may be instructive to write it in components; if, say, $m = 2$, then the $i$-th variable, the $i$-th column of $Y$, is

$$\begin{pmatrix} y_{1i} \\ y_{2i} \\ \dots \\ y_{ni} \end{pmatrix} = \begin{pmatrix} \bar{y}_i \\ \bar{y}_i \\ \dots \\ \bar{y}_i \end{pmatrix} + \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \\ \dots & \\ c_{n1} & c_{n2} \end{pmatrix} \begin{pmatrix} u_{i1} \\ u_{i2} \end{pmatrix} + \begin{pmatrix} d_{1i} \\ d_{2i} \\ \dots \\ d_{ni} \end{pmatrix}$$

that is, $\mathbf{y}_i - \bar{y}_i \mathbf{1} = \mathbf{C}_m \mathbf{u}_i + \mathbf{d}_i$ ($\mathbf{d}_i$ being the columns of $\mathbf{D}$)

and together all $i = 1, 2, \dots, p$ give the matrix form above.

## An attempt at assumptions

Note that although it looks like a regression model, the difference is that *both* $\mathbf{C}_m$ and $\mathbf{U}_m$ are unobserved. Having that many unobserved quantities brings a lot of ambiguity (nonuniqueness of solutions); to alleviate at least some of that, we require

the *columns* of $\mathbf{C}_m$ to have

zero mean and unit standard deviation

are uncorrelated with the columns of $\mathbf{D}$

are uncorrelated among themselves (however: *sometimes* not)

and the *columns* of $\mathbf{D}$ to have zero mean...

...but when we would like to postulate that the columns of $\mathbf{E}$ to be uncorrelated, then in general this will neither be true nor feasible

SO THIS IS NOT A SUCCESS: NOT A WAY TO GO

NOT A SUITABLE MODEL FOR FACTOR ANALYSIS

(Only a perhaps illuminating motivation)

## This is not a model; so what is?

Note: unlike for principal components (and later for canonical correlations), where we did not feel a need for any underlying stochastic model -

in factor analysis we have to start with the corresponding *stochastic*, “population” model instead (that is, we have to introduce underlying distribution from which our data are sampled from, represented by random variables and make assumptions about them) - and then the results of factor analysis will estimates of the parameters of the stochastic model

So “forget” the above, and start to think from scratch in terms of random variables (vectors)

## Independent, identically distributed (iid) sampling

The first part of the model, the independent random sampling paradigm, will be invoked also in many other techniques - it is not exclusive to factor analysis

We say that the data matrix $\mathbf{Y}$ arises via **iid sampling**, if its *lines* can be viewed as outcomes of an independent random vectors (independent between themselves, but not necessarily with independent components) $\mathbf{y}_i$ (or $\mathbf{y}_i^\top$, if you wish), all with the same distribution

- which for notational and other convenience we may sometimes view as the distribution of some generic random vector $\mathbf{y}$, with the same dimension as all $\mathbf{y}_i$

For instance, the componentwise expected value of all $\mathbf{y}_i$ is conveniently expressed via formula $\boldsymbol{\mu} = \mathrm{E}(\mathbf{y})$

For $\mathbf{Y}$ arising by iid sampling, the models for factor analysis come in two varieties, two forms: **predictive (regression) factor model** (called also **orthogonal factor form**, to emphasize the fact that factors are assumed uncorrelated) and the **variance-covariance factor model**.

## Predictive (regression) factor model

It is assumed that $\mathbf{y} - \boldsymbol{\mu} = \mathbf{L}\mathbf{f} + \mathbf{e}$ where

$\mathbf{y}$ is a $p \times 1$ *random* vector

$\mathbf{L}$ is $p \times m$ matrix of *fixed, but unknown* **loadings**

$\mathbf{f}$ is an $m \times 1$ *random* vector of **common factors**

and $\mathbf{e}$ is a $p \times 1$ *random* vector $\mathbf{e}$ of **specific factors**

all random quantities involved have *zero expectation*

$\mathrm{E}(\mathbf{f}) = \mathbf{0}$, $\mathrm{E}(\mathbf{e}) = \mathbf{0}$

and all are *uncorrelated*:

$\mathrm{Cov}(\mathbf{f}, \mathbf{e}) = \mathbf{O}$

the components of $\mathbf{f}$ with the components of $\mathbf{e}$

$\mathrm{Var}(\mathbf{f}) = \mathbf{I}$

the components of $\mathbf{f}$ among themselves - and also have *unit variance*; however, while the components of $\mathbf{e}$ are also uncorrelated among themselves, their variance is not restricted

$\mathrm{Var}(\mathbf{e})$ is a diagonal matrix $\boldsymbol{\Psi}$ (not necessarily equal to $\mathbf{I}$)

## Rotations

Although factor model looks pretty much like a regression model, be aware of the fact that it is essentially different: *both* $\mathbf{f}$ and $\mathbf{L}$ are unobserved here (in regression, we would know $\mathbf{f}$)

Also, the $\mathbf{e}$'s should be viewed as "specific factors" rather than "errors", although the similarity here is close

Having that many unobserved quantities in a model brings a great deal of ambiquity (nonuniqueness), despite the moment restrictions in the assumptions. Inserting $\mathbf{I} = \mathbf{A}\mathbf{A}^\top$, where $\mathbf{A}$ is any orthogonal matrix, in the factor model

$$\mathbf{y} = \boldsymbol{\mu} + \mathbf{L}\mathbf{f} + \mathbf{e} \qquad \text{we obtain}$$

$$\mathbf{y} = \boldsymbol{\mu} + \mathbf{L}\mathbf{f} + \mathbf{e} = \boldsymbol{\mu} + \mathbf{L}\mathbf{I}\mathbf{f} + \mathbf{e} = \boldsymbol{\mu} + (\mathbf{L}\mathbf{A})(\mathbf{A}^\top\mathbf{f}) + \mathbf{e}$$

which yields a new model, equivalent to the previous one, and satisfying all the assumptions - a model with the new factors $\mathbf{A}^\top\mathbf{f}$ and the new loadings $\mathbf{L}\mathbf{A}$

Hence, the model description in factor analysis should be viewed not as a single model, but rather as an equivalence class of models[5]

[5] See also Problem 24

## Rotations: aiding and abetting interpretation

The just demonstrated ambiguity may not be essential when we think just about reducing dimension for predicting **y**; but once interpretation of the common factors is desired, some equivalent versions may be more insightful than the other

Thus, we have seen that once we have *some* solution, some estimates/predictions of the components of the predictive model, other possible versions of the solution can be obtained by **rotation**: a transformation represented by an orthogonal matrix **A**

In factor analysis, a fortuitous choice of **A** may improve the interpretability of the resulting factor model ("the art of rotations")

The choice of **A** can be a result of interpreter's skill and experience; there exist also ways of arriving to suitable rotations algorithmically - the specific proposals are referred to under names like `varimax`, `quartimax`, `equamax`, `parsimax`, `varimin`, `entropy`, `mccammon`,... (see R package `GPArotation`)

## A favorite one: `varimax`

Good rotations are often considered to be those giving as much zeros as possible in the loadings. A popular one known to behave favorably in this respect, a default in many implementations, is the rotation called **varimax**

(*varimax* = *vari*ance *max*imal)

To achieve sparse loadings, the loadings with many zeros, `varimax` maximizes, over all *orthogonal* **A**, the criterion expressed in terms of the elements $\ell_{ij}$ of the matrix **L** of *rotated* loadings

$$\sum_{j=1}^{m}\left(\sum_{i=1}^{p}(\ell_{ij}^2 - \frac{1}{p}\sum_{i=1}^{p}\ell_{ij}^2)^2\right) = \sum_{j=1}^{m}\left(\sum_{i=1}^{p}\ell_{ij}^4 - \frac{1}{p}(\sum_{i=1}^{p}\ell_{ij}^2)^2\right)$$

The maximization is achieved via an iterative algorithm

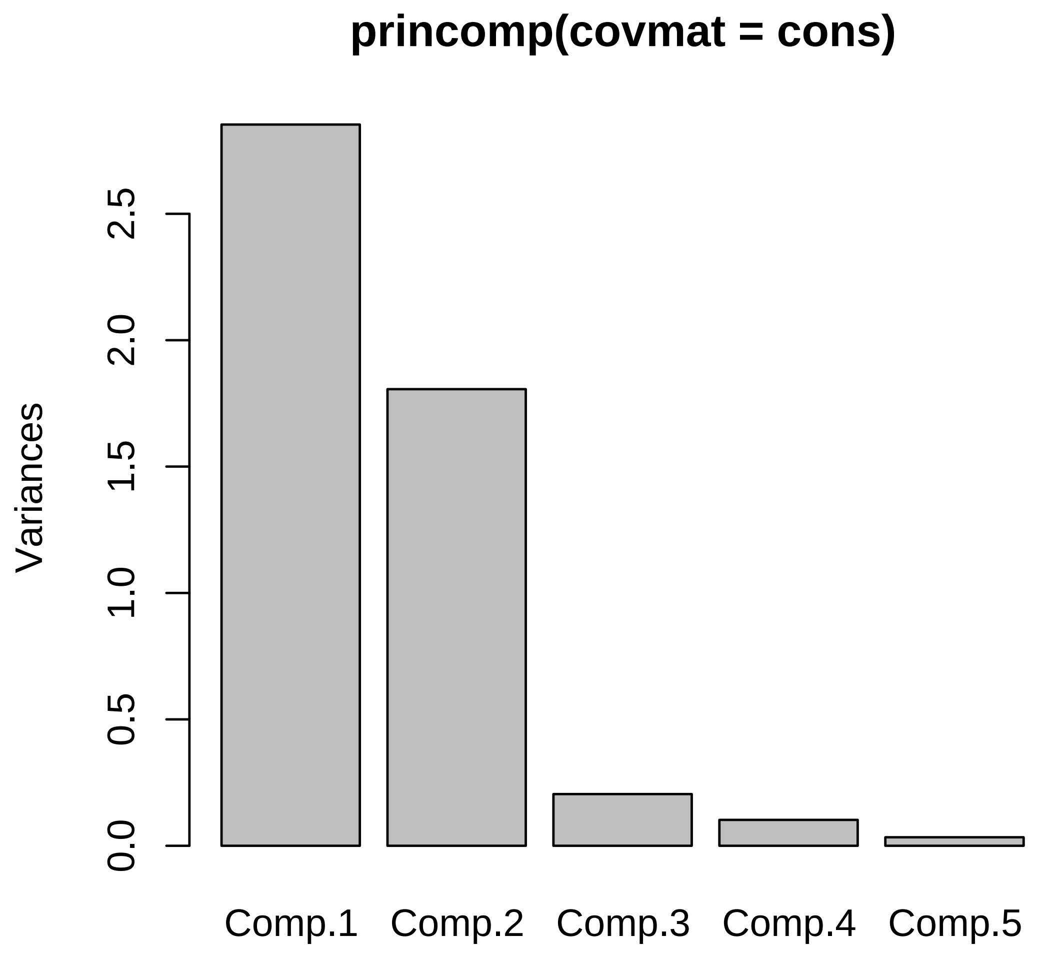
# Consumer preference data: chocolate bars

In a consumer preference study, a random sample of consumers were asked to rate several attributes of a new product. The responses, on a 7-point semantic differential scale, were tabulated and the attribute correlation matrix constructed.

```
> cons
                        Taste Money Flavor Snack Energy
Taste                    1.00  0.02   0.96  0.42   0.01
Good buy for money       0.02  1.00   0.13  0.71   0.85
Flavor                   0.96  0.13   1.00  0.50   0.11
Suitable for snack       0.42  0.71   0.50  1.00   0.79
Provides lots of energy  0.01  0.85   0.11  0.79   1.00

> princomp(covmat=cons)
Standard deviations:
   Comp.1    Comp.2    Comp.3    Comp.4    Comp.5
1.6891094 1.3439987 0.4522060 0.3200148 0.1835141

> plot(princomp(covmat=cons))
```

## Consumer preference data: chocolate bars

First two components summarize data quite well:



## Principal components revisited

As we know only the correlation matrix, the correct computation is:

```
> eigen(cons)$values
[1] 2.85309042 1.80633245 0.20449022 0.10240947 0.03367744
> princomp(covmat=cons)$sdev^2
    Comp.1     Comp.2     Comp.3     Comp.4     Comp.5
2.85309042 1.80633245 0.20449022 0.10240947 0.03367744
```

Other ways may give various dubious (and wrong!) results:

```
> prcomp(cons)$sdev^2
[1] 8.371054e-01 3.307840e-02 5.509727e-03 5.964502e-04 6.841399e-34
> prcomp(cons,scale=T)$sdev^2
[1] 4.533917e+00 4.345533e-01 2.799983e-02 3.530248e-03 6.059843e-33
> princomp(cons,cor=T)$sdev^2
     Comp.1      Comp.2      Comp.3      Comp.4      Comp.5
4.533916642 0.434553284 0.027999826 0.003530248 0.000000000
> eigen(cor(cons))$values
[1] 4.533917e+00 4.345533e-01 2.799983e-02 3.530248e-03 1.201875e-16
> eigen(var(cons))$values
[1]  8.371054e-01  3.307840e-02  5.509727e-03  5.964502e-04 -1.594915e-17
> eigen(var(cons)*4/5)$values
[1] 6.696843e-01 2.646272e-02 4.407781e-03 4.771602e-04 9.421876e-17
> princomp(cons)$sdev^2
      Comp.1       Comp.2       Comp.3       Comp.4       Comp.5
0.6696843389 0.0264627197 0.0044077813 0.0004771602 0.0000000000
```

## Factor analysis of consumer preferences

```
> factanal(cov=cons,factors=2,rotation="none")

Uniquenesses:
[1] 0.028 0.237 0.040 0.168 0.052

Loadings:
     Factor1 Factor2
[1,]  0.976  -0.139
[2,]  0.150   0.860
[3,]  0.979
[4,]  0.535   0.738
[5,]  0.146   0.963
```

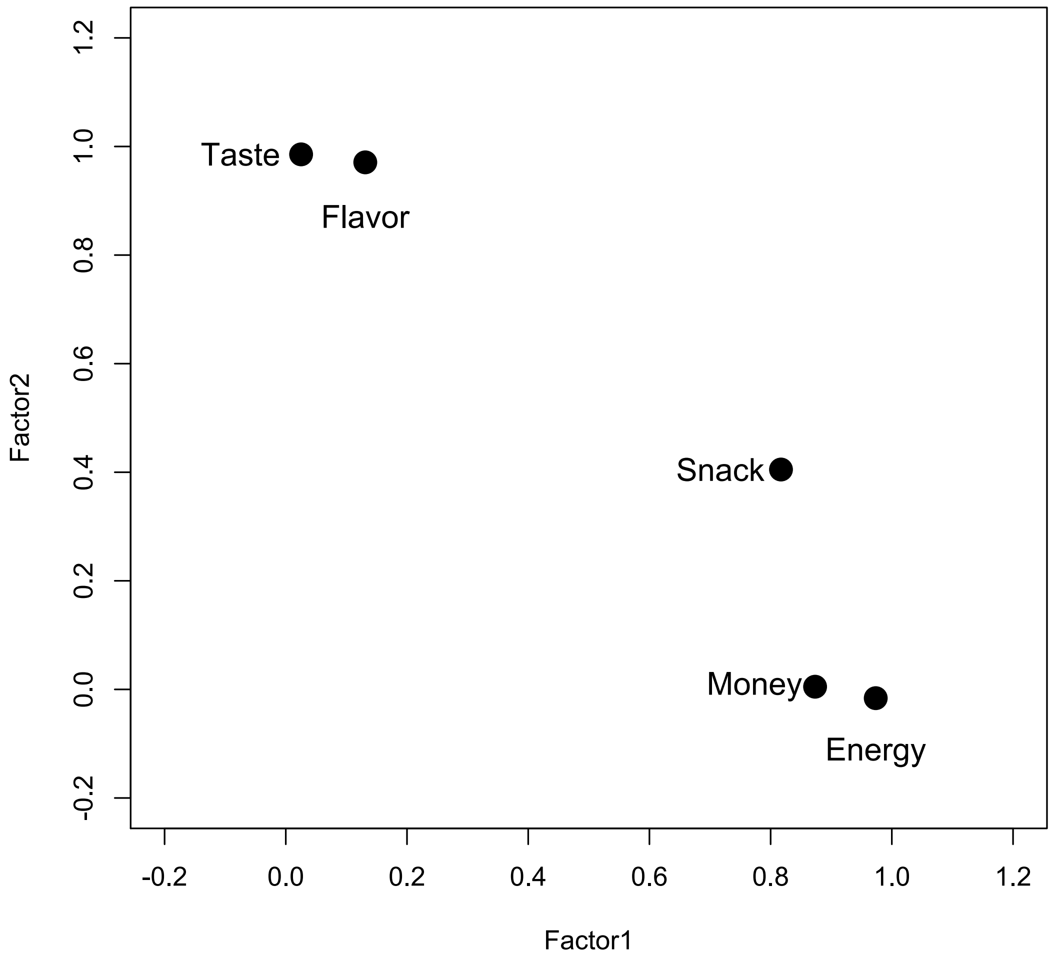
Now, let us try some rotation, to see whether we can transform the result by a suitable rotation to obtain better interpretation

## Varimax

```
> fcp = factanal(cov=cons,factors=2,rotation="varimax")

Uniquenesses:
[1] 0.028 0.237 0.040 0.168 0.052

Loadings:
     Factor1 Factor2
[1,]          0.985
[2,]  0.873
[3,]  0.131   0.971
[4,]  0.817   0.405
[5,]  0.973
```

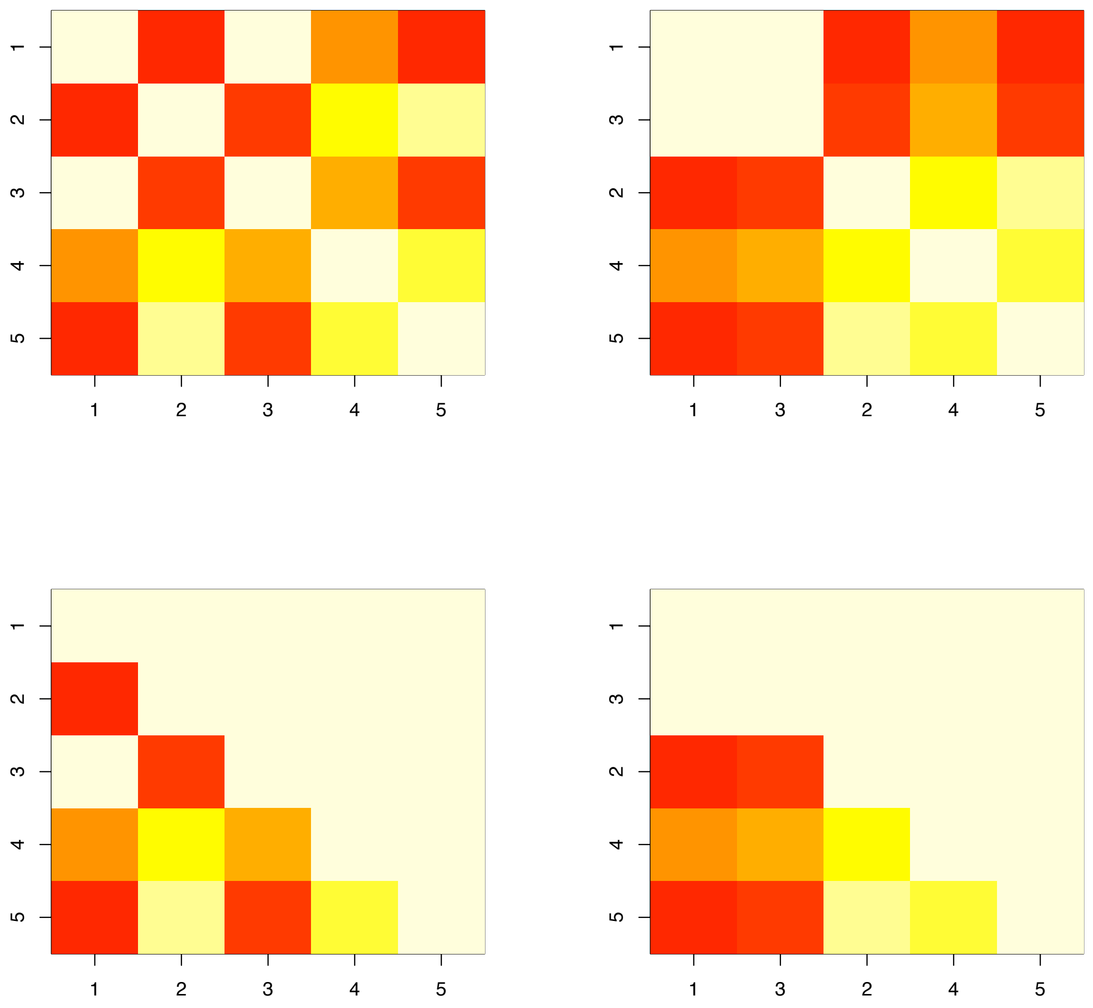
# How about a plot?

```
> plot(loadings(fcp),xlim=c(-0.2,1.2),ylim=c(-0.2,1.2),cex=2,pch=16)
> text(loadings(fcp)-c(0.1,0.1,0,0.1,0,0,0,0.1,0,0.1),
+ labels=dimnames(cons)[[2]],cex=1.2)
```



## Plotting of variance-covariance matrices

We may plot also original and permuted (1,3,2,4,5) correlation matrix: (hint: use `image()`)



## In fact, it is all about variance-covariance matrix

The orthogonal factor analysis model in the regression form has consequences for the variance-covariance matrix of the random vector with underlying distribution. Recall the formula

$$\mathrm{Var}(\mathbf{y}) = \mathrm{E}[(\mathbf{y}-\boldsymbol{\mu})(\mathbf{y}-\boldsymbol{\mu})^\top)]$$

The factor model says that $\mathbf{y}-\boldsymbol{\mu} = \mathbf{Lf}+\mathbf{e}$. This yields

$$\begin{aligned}\mathrm{Var}(\mathbf{y}) &= \mathrm{E}[(\mathbf{Lf}+\mathbf{e})(\mathbf{f}^\top\mathbf{L}^\top+\mathbf{e}^\top)]\\ &= \mathbf{L}\,\mathrm{E}(\mathbf{ff}^\top)\mathbf{L}^\top + \mathbf{L}\,\mathrm{E}(\mathbf{fe}^\top) + \mathrm{E}(\mathbf{ef}^\top)\mathbf{L}^\top + \mathrm{E}(\mathbf{ee}^\top)\end{aligned}$$

and, because $\mathrm{E}(\mathbf{f}) = 0$, $\mathrm{E}(\mathbf{e}) = 0$

$$\begin{aligned} &= \mathbf{L}\,\mathrm{Var}(\mathbf{f})\mathbf{L}^\top + \mathbf{L}\,\mathrm{Cov}(\mathbf{f},\mathbf{e}) + \mathrm{Cov}(\mathbf{e},\mathbf{f})\mathbf{L}^\top + \mathrm{Var}(\mathbf{e})\\ &= \mathbf{L}\,\mathrm{Var}(\mathbf{f})\mathbf{L}^\top + \boldsymbol{\Psi} = \mathbf{LL}^\top + \boldsymbol{\Psi}\end{aligned}$$

using that $\mathrm{Var}(\mathbf{f}) = \mathbf{I}$ and $\mathrm{Cov}(\mathbf{f},\mathbf{e})$ is a zero matrix

In a similar way, we obtain that $\mathrm{Cov}(\mathbf{y},\mathbf{f}) = \mathbf{L}$[6]

[6] Problem 25

## Variance-covariance factor model

Thus, the orthogonal factor model in the regression form implies the following (orthogonal) factor model for the variance-covariance matrix:

The components $\mathbf{L}$, $\mathbf{f}$, and $\mathbf{e}$ are the same, assumed to have the same stochastic or deterministic character as in the predictive regression model

and it is assumed that

$$\mathrm{Var}(\mathbf{y}) = \mathbf{L}\mathbf{L}^{\mathsf{T}} + \boldsymbol{\Psi}$$

and

$$\mathrm{Cov}(\mathbf{y}, \mathbf{f}) = \mathbf{L}$$

Also this form of factor model remains unchanged under the orthogonal rotation of factors, when $\mathbf{L}$ is replaced by $\mathbf{LA}$, for any orthogonal matrix $\mathbf{A}$

## Communalities and uniquenesses

In terms of the elements $\ell_{ij}$ of $\mathbf{L}$, we obtain

$$\mathrm{Var}(y_i) = \ell_{i1}^2 + \cdots + \ell_{im}^2 + \psi_i,$$

$$\mathrm{Cov}(y_i, y_j) = \ell_{i1}\ell_{j1} + \cdots + \ell_{im}\ell_{jm}, \qquad \mathrm{Cov}(y_i, f_j) = \ell_{ij}.$$

Diagonal components of $\mathbf{LL}^\top$, those that equal to the sum of $\ell_{ij}^2$ for $j = 1, 2, \ldots, m$, are called **communalities**

Note that communalities are independent of rotations

Diagonal elements $\psi_i$ of $\boldsymbol{\Psi}$ are called **uniquenesses**, or *specific variances*

Note: the factor analysis model for the variance-covariance matrix is *not* equivalent to the predictive form; it follows from it, but its scope is more narrow. It is sufficient to adopt if we are interested only in factors and their interpretation; it does not provide the guidelines for the prediction of the individual factor scores

## Estimation in factor analysis

Loadings and uniquenessess are now estimated as parameters - that is, for the whole sample; such an estimation may require only the knowledge of var$(\mathbf{Y})$. Factors may be then estimated from $\mathbf{Y}$ as unobserved quantities specific for the $i$-th item.

There are two widely used methods: the first one[7] estimates loadings from principal components, obtaining the estimates through **low-rank approximation** $\widehat{\mathbf{L}}\widehat{\mathbf{L}}^\top$ of the sample variance-covariance matrix $\mathbf{S_Y}$ - or, in practice, rather of the correlation matrix $\mathbf{R_Y}$. The estimates of uniquenesses, of the diagonal elements of $\boldsymbol{\Psi}$ are then obtained as the *diagonal elements* of $\mathbf{S_Y} - \widehat{\mathbf{L}}\widehat{\mathbf{L}}^\top$; non-diagonal elements are dismissed

*If estimates are obtained in this way, they are all nonnegative*

A modification of the above scheme is possible too: uniquenesses estimated first, then the diagonal matrix formed by them subtracted from $\mathbf{S_Y}$; the resulting matrix then low-rank approximated as above. New estimates of uniquenesses can be obtained in this way and the whole process can be iterated

[7]See Problem 26

# Low-rank approximation

The problem is defined as follows: given a matrix $\mathbf{A}$, we are looking for a matrix $\mathbf{B}$, with rank not greater than $m$, which minimizes the Euclidean (Frobenius, Hilbert-Schmidt) distance of $\mathbf{A}$ and $\mathbf{B}$.

By its definition, the square of the Euclidean (Hilbert-Schmidt) distance, $d^2(\mathbf{A}, \mathbf{B})$, is just the sum of squares of all element-wise differences. In matrix notation, it can be expressed as

$$\operatorname{tr}(\mathbf{A}-\mathbf{B})(\mathbf{A}-\mathbf{B})^\top = \operatorname{tr}(\mathbf{A}-\mathbf{B})^\top(\mathbf{A}-\mathbf{B})$$

(if both matrices are symmetric, as is our case, then the transposition sign can be dropped)

For symmetric, nonnegative definite matrices, the approximation is easily found via the eigenvalue decomposition. If $\mathbf{A} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top$, with $\mathbf{U}$ orthogonal, $\boldsymbol{\Lambda}$ diagonal, and $m$ is the desired approximation rank, the we put $\mathbf{B} = \mathbf{U}\boldsymbol{\Lambda}_m\mathbf{U}^\top$, where $\boldsymbol{\Lambda}_m$ is the matrix obtained from $\boldsymbol{\Lambda}$ by taking its $m$ largest diagonal elements, $m$ largest eigenvalues (all eigenvalues should be $\geqslant 0$), and replacing the rest of them by 0. As

$$\mathbf{A} = \mathbf{U}\boldsymbol{\Lambda}_m^{1/2}\boldsymbol{\Lambda}_m^{1/2}\mathbf{U}^\top = \mathbf{U}\boldsymbol{\Lambda}_m^{1/2}(\mathbf{U}\boldsymbol{\Lambda}_m^{1/2})^\top$$

we can also see that how $\mathbf{A}$ can be written as $\widehat{\mathbf{L}}\widehat{\mathbf{L}}^\top$.

## Estimation: maximum likelihood

The method based on low-rank approximation is considered somewhat historic - the method predominantly used these days is **maximum** likelihood. It is based on the factor model and normality assumption about $\mathbf{y}$

Its advantage is *yielding the same loadings even from rescaled data*, consistently with the stochastic factor model; but, on the other hand, it may introduce so-called **Heywood cases**, estimates of uniquenesses that are 0 or negative.

The latter are then usually adjusted: either directly to 0, or in some other way: some implementations constrain uniquenesses to be $\geqslant$ than some preselected small number

Once loadings are estimated, then they are usually also rotated, to facilitate their interpretation.

## Maximum likelihood and testing for $m$

Maximum likelihood estimates are particularly suitable testing with the aim to assess $m$, the number of factors.

The null hypothesis $H_0$ that the variance-covariance matrix has the form $\mathrm{Var}(\mathbf{y}) = \mathbf{L}\mathbf{L}^\mathsf{T} + \boldsymbol{\Psi}$ is tested again a general $\mathrm{Var}(\mathbf{y})$ alternative using likelihood ratio test, using the ratio of the maximized likelihood under $m$ factors and the maximized likelihood in general, that is, under $p$ factors. The $-2\log$ of this ratio is

$$n \log\left(\frac{\det(\widehat{\mathbf{L}}\widehat{\mathbf{L}}^\mathsf{T} + \widehat{\boldsymbol{\Psi}})}{\det(\frac{n-1}{n}\mathbf{S}_\mathbf{Y})}\right)$$

where $\widehat{\mathbf{L}}$ and $\widehat{\boldsymbol{\Psi}}$ are maximum likelihood estimates under the factor model, and $\frac{n-1}{n}\mathbf{S}_\mathbf{Y}$ is the maximum likelihood estimate of unrestricted variance-covariance matrix $\mathrm{Var}(\mathbf{y})$ - all under normality assumption

The test statistic has approximately $\chi^2_{\frac{1}{2}[(p-m)^2-p-m]}$ distribution; the approximation is improved (Bartlett) by replacing $n$ by $(n-1-(2p+4m+5)/6)$. The null hypothesis that $m$ factors are sufficient is rejected when the statistic is large.

## Oblique rotations

Orthogonal rotations were obtained by inserting $\mathbf{I} = \mathbf{A}\mathbf{A}^\top$ into the regression form of a factor model; inserting $\mathbf{I}$ in the form $\mathbf{A}\mathbf{A}^{-1}$, where $\mathbf{A}$ is not necessarily an orthogonal matrix, but merely an invertible one results in a so-called **oblique rotation**

Such a move many consider controversial: while it preserves the predictive form with the new loadings $\mathbf{LA}$, the new factors $\mathbf{A}^{-1}\mathbf{f}$ are not only no longer uncorrelated - and by the analogy with uncorrelated principal components, it would be also natural to assume that common factors are uncorrelated - but moreover, the rotated factors do not preserve the variance-covariance factor model (unless the rotation matrix is symmetric, which is not assumed, and is typically not the case; most of the orthogonal matrices used for rotations are not symmetric)

Thus, after general oblique rotation, the resulting common factors are in general correlated - but, the assumption of uncorrelatedness of factors is relaxed in a hope of possibly more unequivocal ("clear-cut") interpretation of factors

Note that oblique factor loadings are estimated in the same way as the orthogonal factors; it is non-orthogonal rotation, requiring only that $\mathbf{A}$ has to be invertible, which makes them oblique

## Promax: the varimax of oblique rotations

The algorithmic methods to obtain interesting oblique rotations come also under various names, like `promax`, `oblimin`, `quartimin`, `oblimax`, `simplimax`, ... (see the R package `GPArotation` again).

A prominent (and often preferred) method, a method with a position in oblique rotations similar to that of `varimax` in the orthogonal rotations, is called **promax**

The method (depends on $s$, typical values of $s$ are $s = 1, 2, 3, 4$) proceeds in the following steps:

1. **L** (orthogonally) rotated by varimax (by an orthogonal matrix **T**); the rotated **LT** becomes new **L**

2. **Q** with the same dimension as **L**, $q_{ij} = |\ell_{ij}^{s-1}|\ell_{ij}$; the nonzero elements are increased or decreased depending on whether they are $> 1$ or $< 1$

3. **LT** is sought (**T** not necessarily orthogonal now) that best approximates **Q** in the Euclidean (Hilbert-Schmidt) metric. The expected effect is to decrease small and increase large loadings

## Oblique rotation for chocolate bars

```
> factanal(cov=cons,factors=2,rotation="promax")
Loadings:
     Factor1 Factor2
[1,]          1.004
[2,]  0.892  -0.103
[3,]          0.975
[4,]  0.786   0.313
[5,]  0.997  -0.137
```

## Estimation of scores: prediction

As mentioned above, loadings and uniqueness can be estimated from the variance-covariance (or correlation) matrix alone. Once the $p \times m$ matrix $\mathbf{L}$ of loadings is estimated, the estimates become "known" regressors in the regression form of the factor model

*If full data are available*, we can also estimate (predict!) factor scores for each given $i$: that is, estimate $\mathbf{f}_i$ under the assumption that it is sampled with the same distribution as $\mathbf{f}$.

There are two ways to do it:

## Bartlett scores

For fixed $i$, the sample version for $i$-th observation given by the factor model is

$$\mathbf{y}_i - \boldsymbol{\mu} = \mathbf{L}\mathbf{f}_i + \mathbf{e}_i$$

Once we "know" $\mathbf{L}$ - that is, we plug in its estimate $\widehat{\mathbf{L}}$ - the estimation of $\mathbf{f}_i$ can be done by the least squares method, as in any standard linear model. Given, however, that the variances of the "errors" are not equal - the matrix $\boldsymbol{\Psi}$ is not diagonal - the usually used estimate is that by the weighted least squares

$$\widehat{\mathbf{f}}_i = (\widehat{\mathbf{L}}\widehat{\boldsymbol{\Psi}}^{-1}\widehat{\mathbf{L}})^{-1}\widehat{\mathbf{L}}^{\top}\widehat{\boldsymbol{\Psi}}^{-1}(\mathbf{y}_i - \bar{\mathbf{y}})$$

where $\widehat{\boldsymbol{\Psi}}$ is the estimate of $\boldsymbol{\Psi}$ obtained by one of the methods described above

The formula is the standard regression estimate, except for the fact that $\boldsymbol{\mu}$ is estimated by

$$\bar{\mathbf{y}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{y}_i.$$

## Thomson (regression) scores

As both $\mathbf{y}_i$ and $\mathbf{f}_i$ are stochastic, we may also regress $\mathbf{f}_i$ on $\mathbf{y}_i$ in a stochastic way (recall that the mean of $\mathbf{f}_i$ is zero). That is, they having the same distribution as $\mathbf{y}$ and $\mathbf{f}$, we first note that $\mathbf{U}$ minimizing

$$\mathrm{E}(\|\mathbf{f} - \mathbf{U}\mathbf{y}\|^2)$$

is (abusing slightly the notation)

$$\mathbf{U} = \mathrm{Cov}(\mathbf{f}, \mathbf{y})(\mathrm{Var}(\mathbf{y})^{-1} = \mathbf{L}^\top(\mathrm{Var}(\mathbf{y}))^{-1}$$

yelding a predictive equation, for every fixed $i$,

$$\mathbf{f}_i = \mathbf{U}_i(\mathbf{y}_i - \boldsymbol{\mu}) = \mathbf{L}^\top(\mathrm{Var}(\mathbf{y}))^{-1}(\mathbf{y_i} - \boldsymbol{\mu})$$

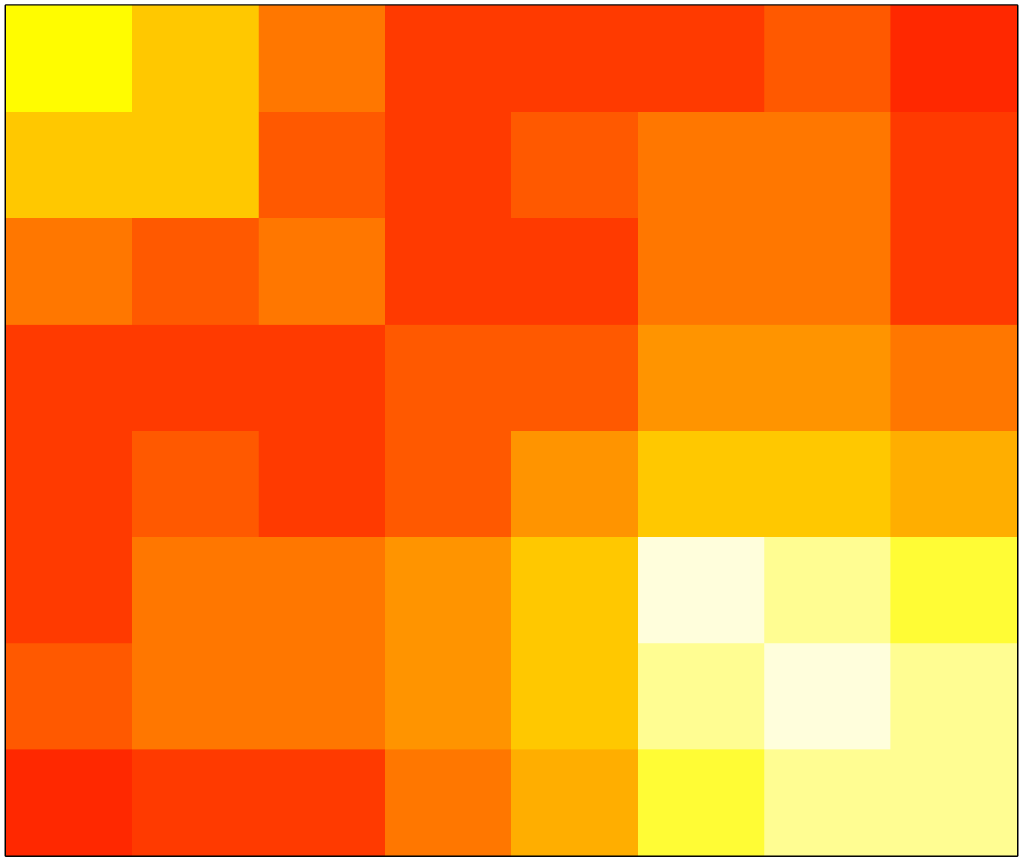
The Thomson scores are then obtained by replacing $\mathbf{L}$, Var($\mathbf{y}$), and $\boldsymbol{\mu}$ respectively by their corresponding estimates $\widehat{\mathbf{L}}$, $\mathbf{S} = \mathrm{var}(\mathbf{Y})$, and $\bar{\mathbf{y}}$

$$\widehat{\mathbf{f}}_i = \widehat{\mathbf{L}}^\top \mathbf{S}^{-1}(\mathbf{y}_i - \bar{\mathbf{y}})$$

## Track velocity records - men: variance matrix

```
     X1    X2    X4    X8    X15   X50   X100  M
X1   0.090 0.080 0.064 0.056 0.056 0.058 0.061 0.048
X2   0.080 0.083 0.063 0.058 0.060 0.064 0.065 0.054
X4   0.064 0.063 0.067 0.057 0.058 0.064 0.066 0.058
X8   0.056 0.058 0.057 0.063 0.061 0.069 0.071 0.064
X15  0.056 0.060 0.058 0.061 0.072 0.080 0.082 0.074
X50  0.058 0.064 0.064 0.069 0.080 0.100 0.100 0.096
X100 0.061 0.065 0.066 0.071 0.082 0.100 0.110 0.099
M    0.048 0.054 0.058 0.064 0.074 0.096 0.099 0.100
```

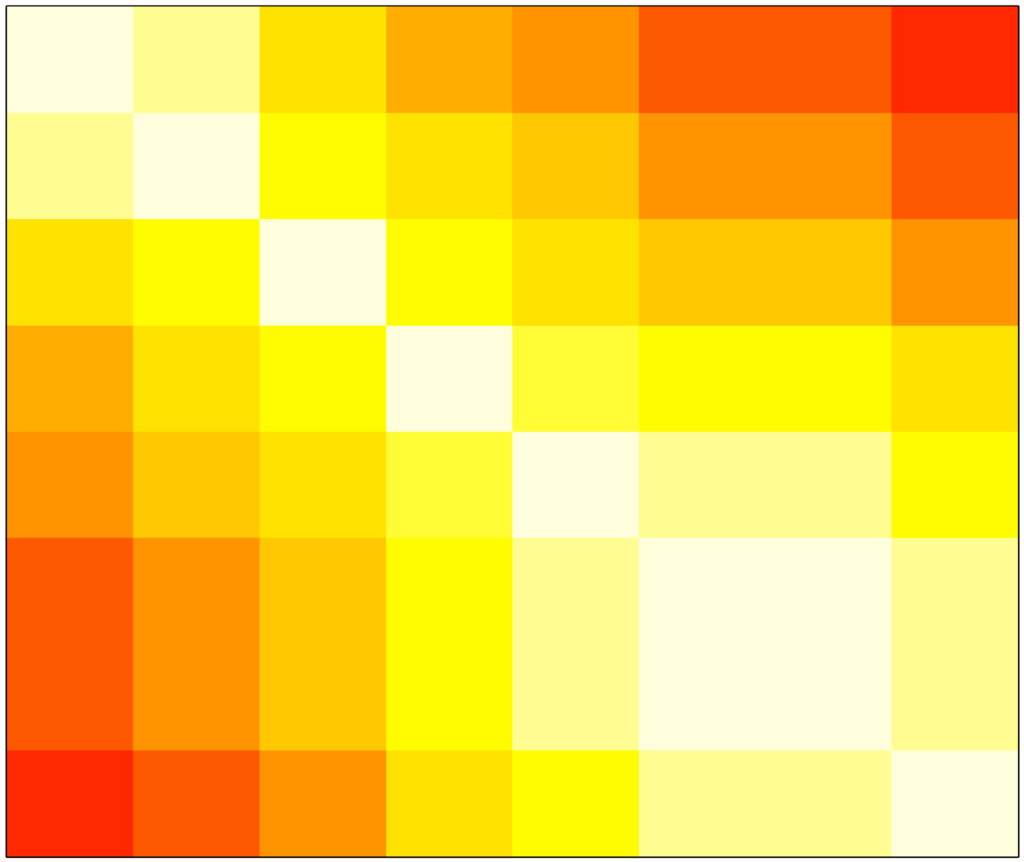
## In colors



# Track velocity records - men: correlation matrix

```
       X1   X2   X4   X8  X15  X50 X100    M
X1   1.00 0.92 0.83 0.75 0.69 0.60 0.61 0.50
X2   0.92 1.00 0.85 0.80 0.77 0.69 0.69 0.59
X4   0.83 0.85 1.00 0.87 0.83 0.77 0.78 0.70
X8   0.75 0.80 0.87 1.00 0.91 0.85 0.86 0.80
X15  0.69 0.77 0.83 0.91 1.00 0.93 0.93 0.86
X50  0.60 0.69 0.77 0.85 0.93 1.00 0.97 0.93
X100 0.61 0.69 0.78 0.86 0.93 0.97 1.00 0.94
M    0.50 0.59 0.70 0.80 0.86 0.93 0.94 1.00
```

## Again in colors



## Track records velocity - men: factor analysis

```
> factanal(trackmen,factors=2,rotation="varimax")
...
Uniquenesses:
   s1    s2    s4    m8   m15   m50  m100  m421
0.081 0.076 0.151 0.135 0.082 0.034 0.018 0.086

Loadings:
     Factor1 Factor2
s1   0.291   0.914
s2   0.382   0.882
s4   0.543   0.744
m8   0.691   0.622
m15  0.799   0.530
m50  0.901   0.394
m100 0.907   0.399
m421 0.915   0.278
...
Test of the hypothesis that 2 factors are sufficient.
The chi square statistic is 16.36 on 13 degrees of freedom.
The p-value is 0.23
```

Note: the p-value indicates that two factors are enough

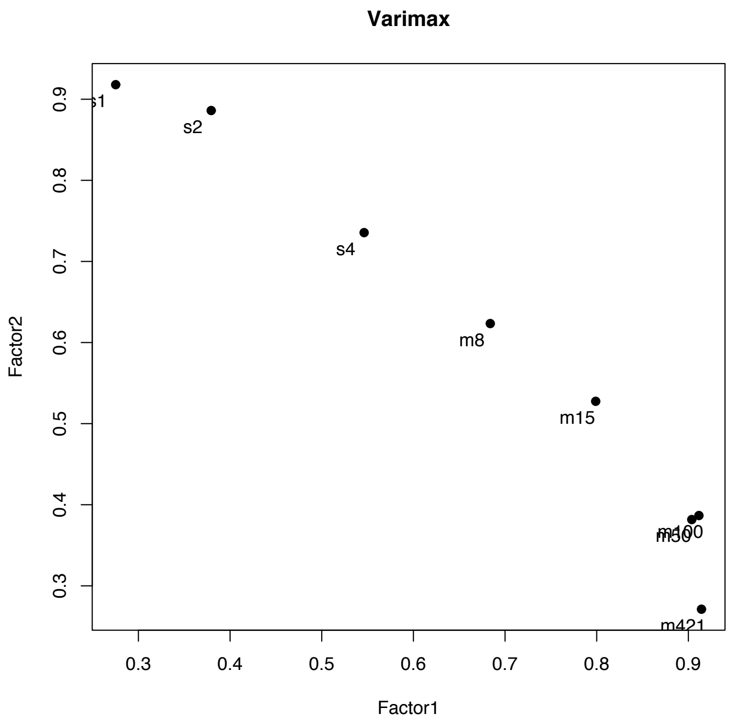
## For comparison: 1 factor

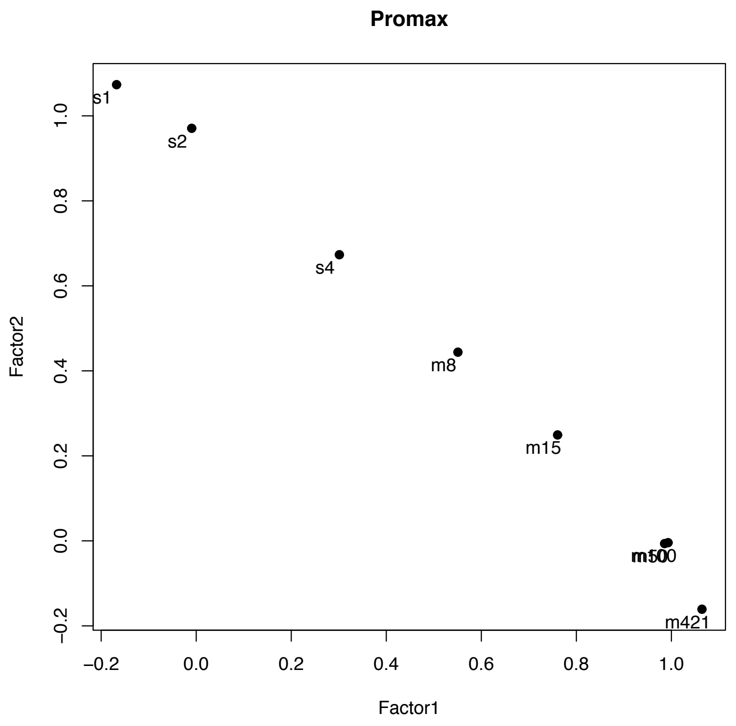
```
> factanal(trackmen,factors=1,rotation="varimax")
...
Uniquenesses:
   s1    s2    s4    m8   m15   m50  m100  m421
0.556 0.461 0.332 0.196 0.094 0.036 0.025 0.118

Loadings:
     Factor1
s1   0.666
s2   0.734
s4   0.817
m8   0.897
m15  0.952
m50  0.982
m100 0.988
m421 0.939
...
Test of the hypothesis that 1 factor is sufficient.
The chi square statistic is 156.37 on 20 degrees of freedom.
The p-value is 3.77e-23
```

## Track records velocity - men: oblique factors

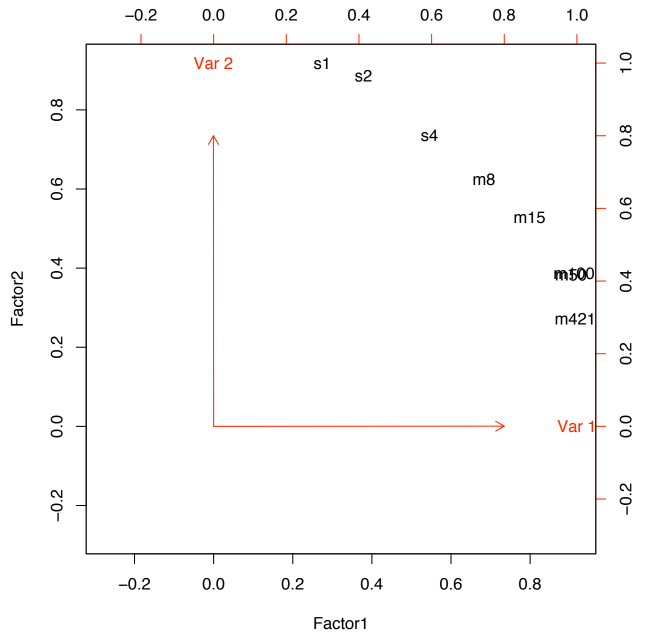
```
> factanal(trackmen,factors=2,rotation="promax")
...
Uniquenesses:
   s1    s2    s4    m8   m15   m50  m100  m421
0.081 0.076 0.151 0.135 0.082 0.034 0.018 0.086
...
     Factor1 Factor2
s1   -0.153   1.066
s2            0.970
s4    0.288   0.689
m8    0.560   0.437
m15   0.760   0.249
m50   0.979
m100  0.985
m421  1.067  -0.160

Factor Correlations:
        Factor1 Factor2
Factor1   1.000   0.735
Factor2   0.735   1.000

Test of the hypothesis that 2 factors are sufficient.
The chi square statistic is 16.36 on 13 degrees of freedom.
The p-value is 0.23
```

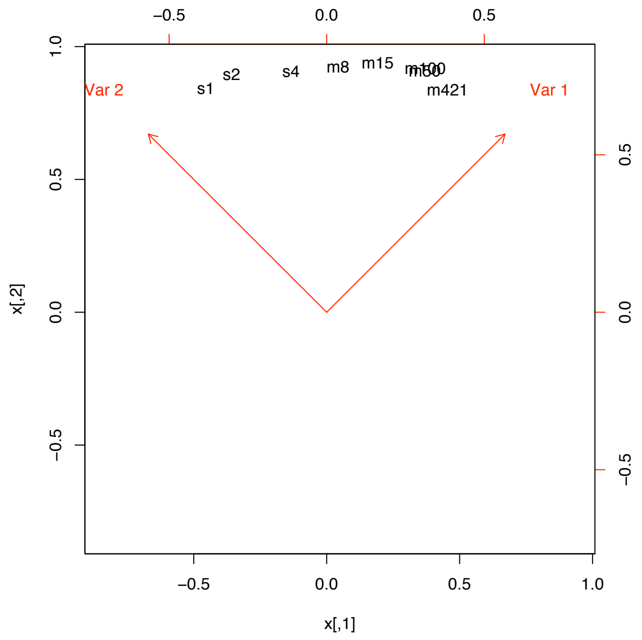
## Track records velocity - men

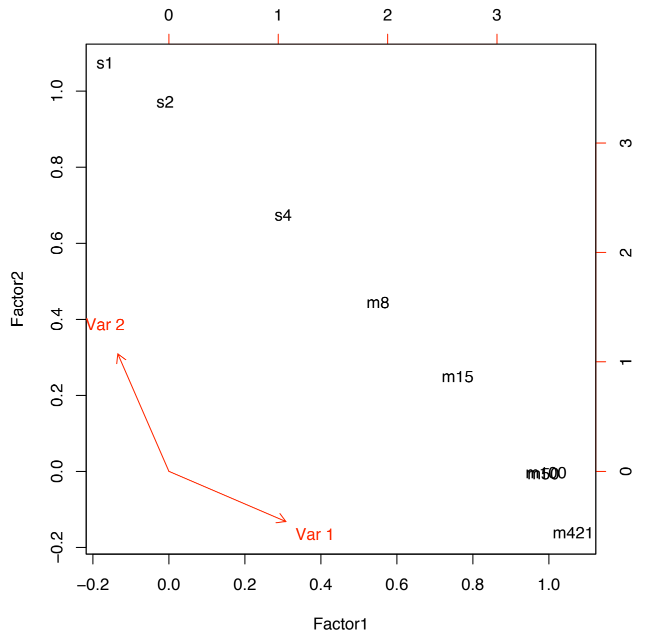


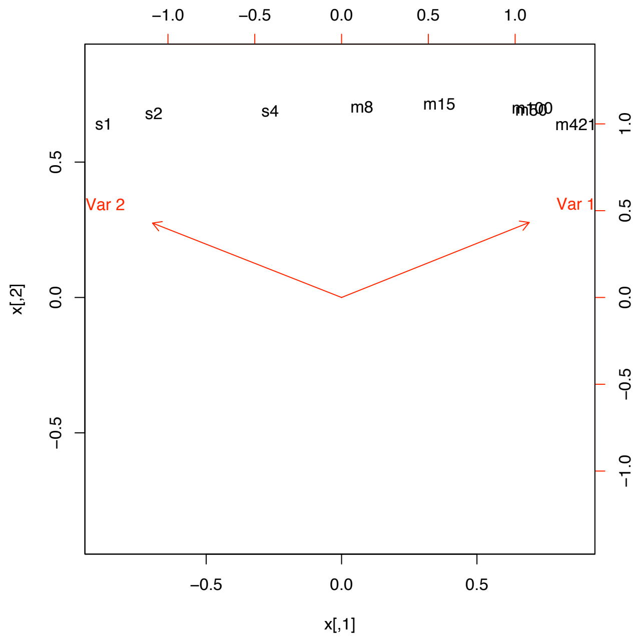


# Track records velocity men - more rotations









## Track records velocity - women: factor analysis

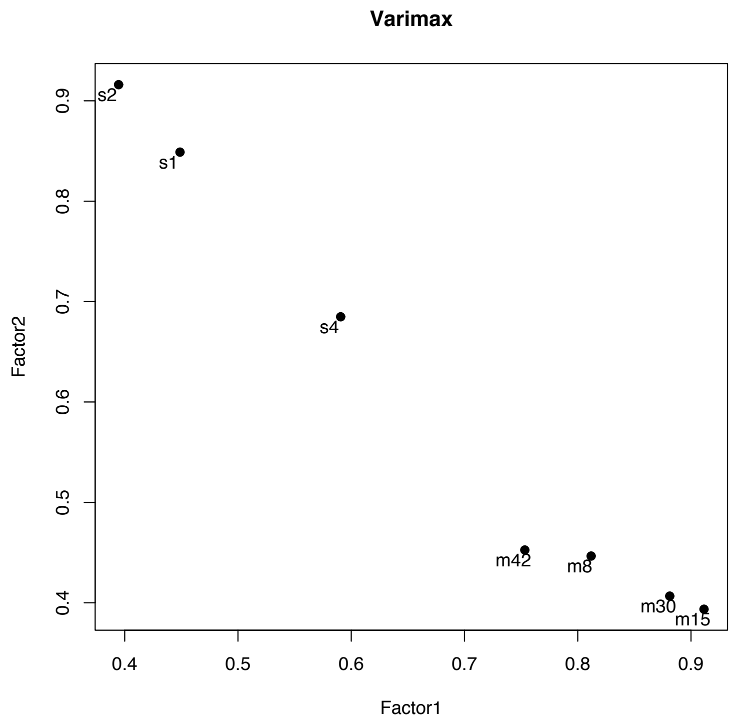
```
factanal(x = welo, factors = 2, rotation = "varimax")

Uniquenesses:
   s1    s2    s4    m8   m15   m30  m421
0.078 0.005 0.182 0.142 0.014 0.058 0.228

Loadings:
     Factor1 Factor2
s1   0.449   0.849
s2   0.395   0.916
s4   0.591   0.685
m8   0.812   0.447
m15  0.912   0.394
m30  0.881   0.407
m421 0.753   0.453

Test of the hypothesis that 2 factors are sufficient.
The chi square statistic is 50.17 on 8 degrees of freedom.
The p-value is 3.78e-08
```

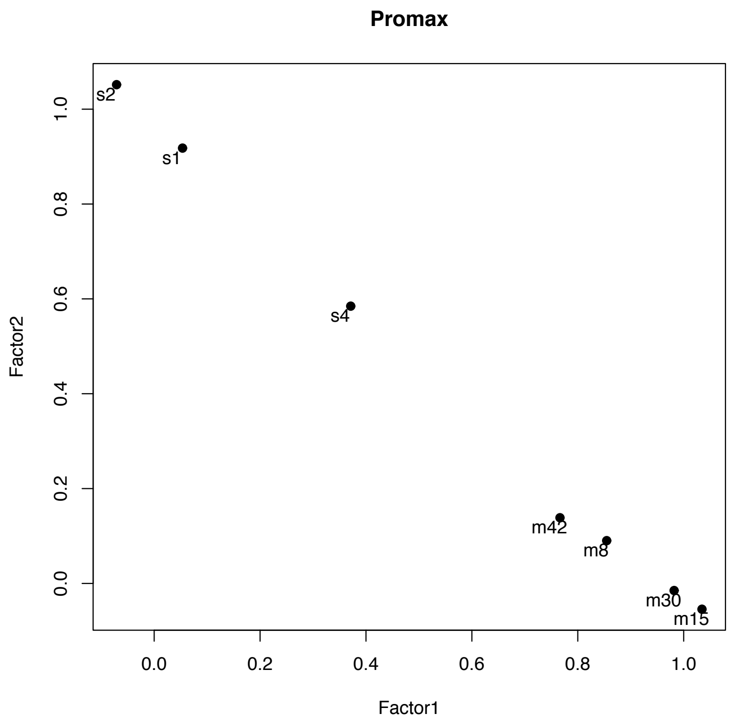
## Track records velocity - women: oblique factors

```
factanal(x = welo, factors = 2, rotation = "promax")

Uniquenesses:
   s1    s2    s4    m8   m15   m30  m421
0.078 0.005 0.182 0.142 0.014 0.058 0.228

Loadings:
     Factor1 Factor2
s1            0.918
s2            1.052
s4    0.371   0.585
m8    0.855
m15   1.035
m30   0.982
m421  0.766   0.139

Test of the hypothesis that 2 factors are sufficient.
The chi square statistic is 50.17 on 8 degrees of freedom.
The p-value is 3.78e-08
```

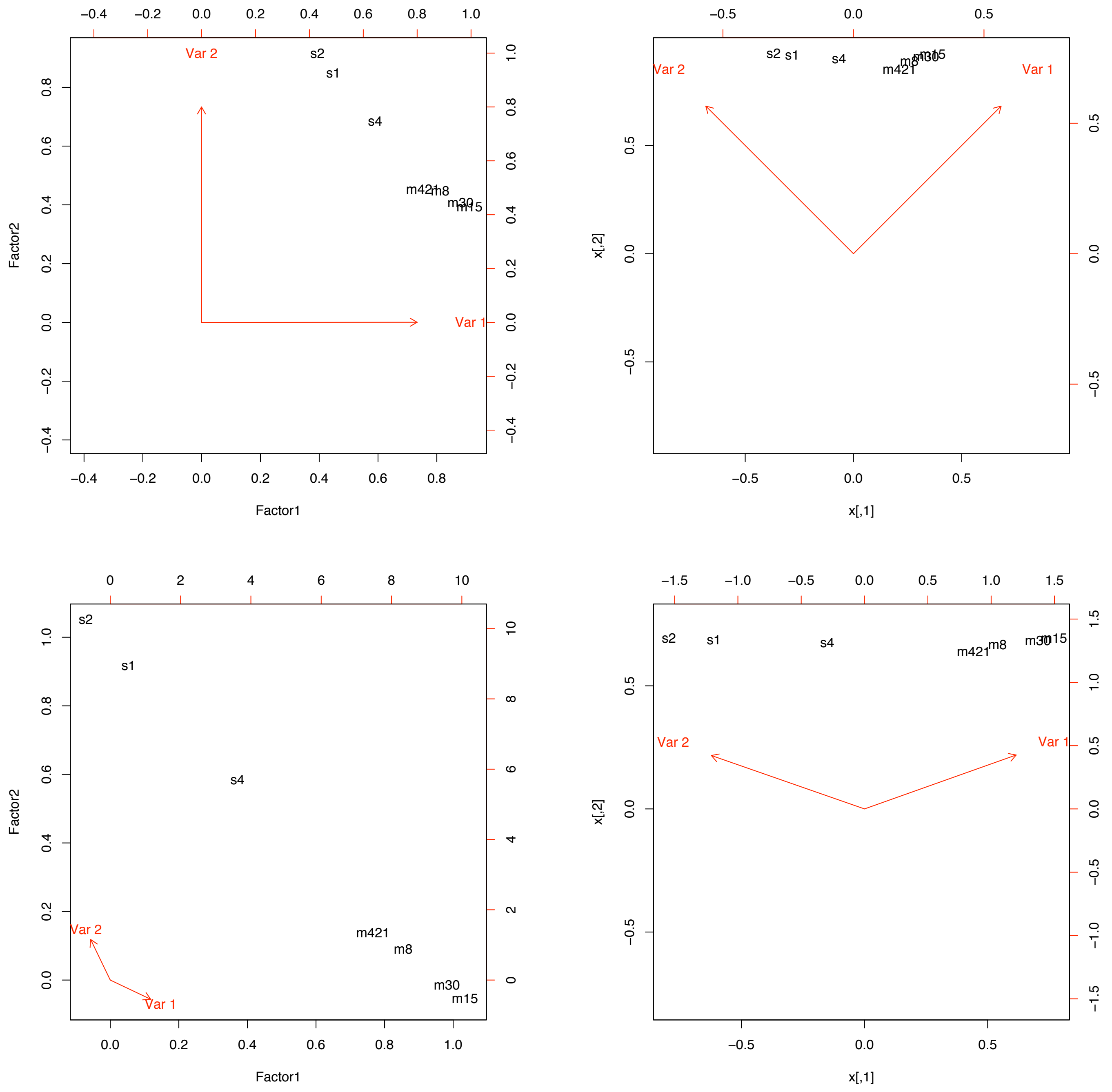
Note: p-values indicate that two factors are *not* enough.

# Track records velocity - women





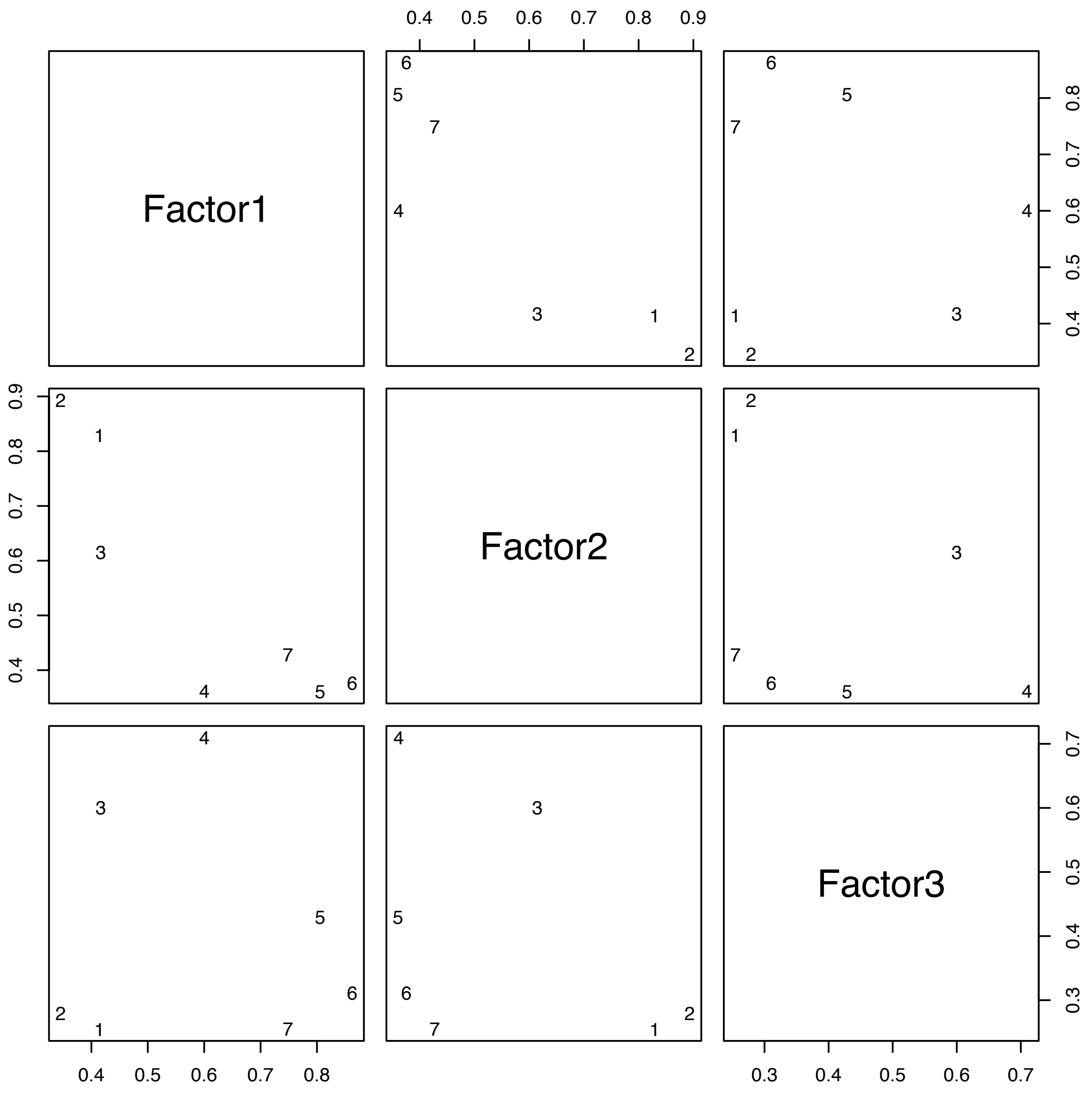
# Track records velocity women - more rotations



# Track records velocity women - 3 factors

```
factanal(x = welo, factors = 3, rotation = "varimax")

Uniquenesses:
   s1    s2    s4    m8   m15   m30  m421
0.076 0.005 0.088 0.005 0.037 0.018 0.192

Loadings:
     Factor1 Factor2 Factor3
s1   0.414   0.829   0.255
s2   0.345   0.893   0.279
s4   0.417   0.615   0.600
m8   0.600   0.362   0.710
m15  0.806   0.360   0.429
m30  0.863   0.376   0.311
m421 0.749   0.428   0.255

Test of the hypothesis that 3 factors are sufficient.
The chi square statistic is 4.65 on 3 degrees of freedom.
The p-value is 0.199
```

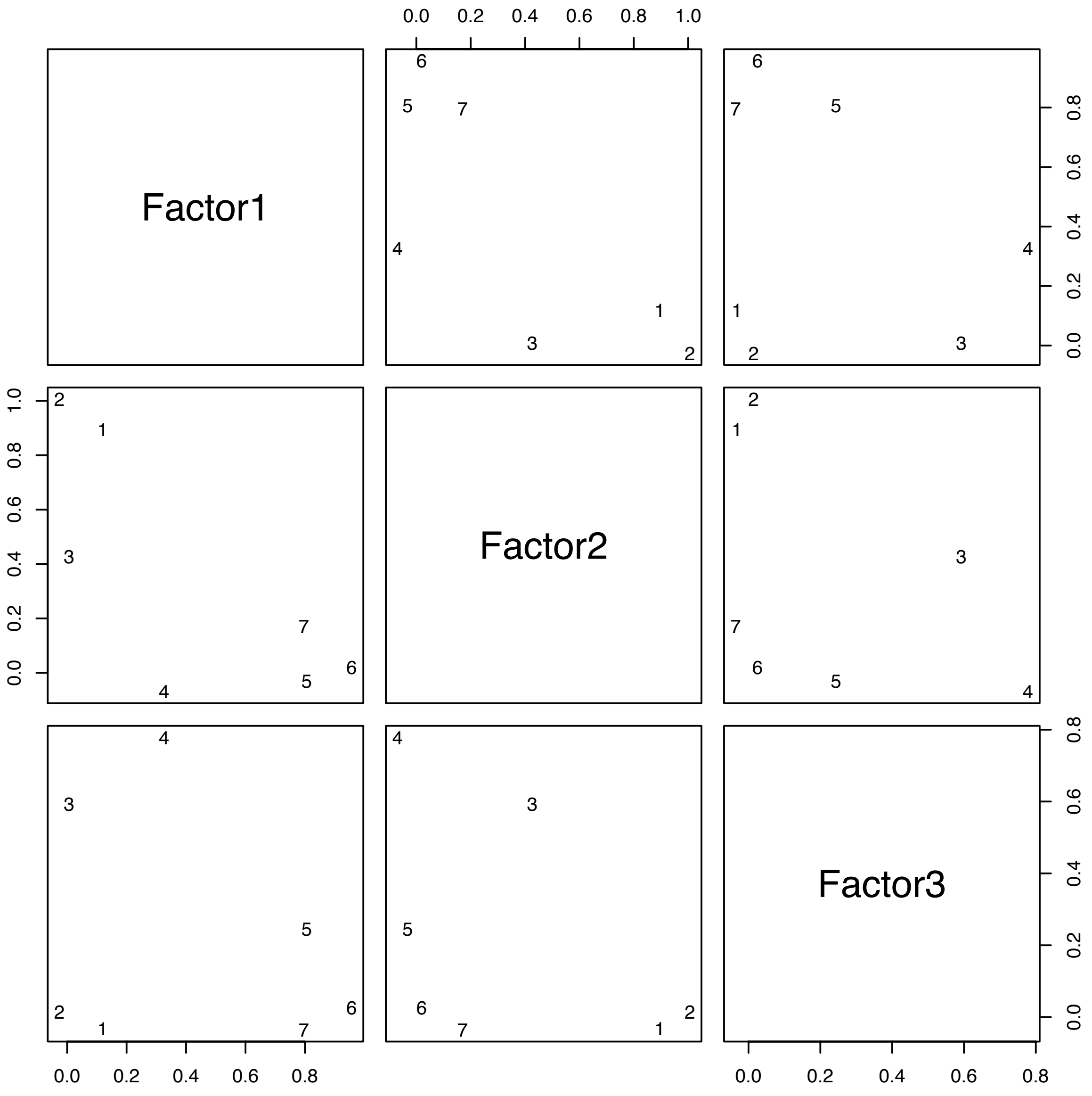
## Plotting those 3 factors



## Track records velocity women - 3 oblique factors

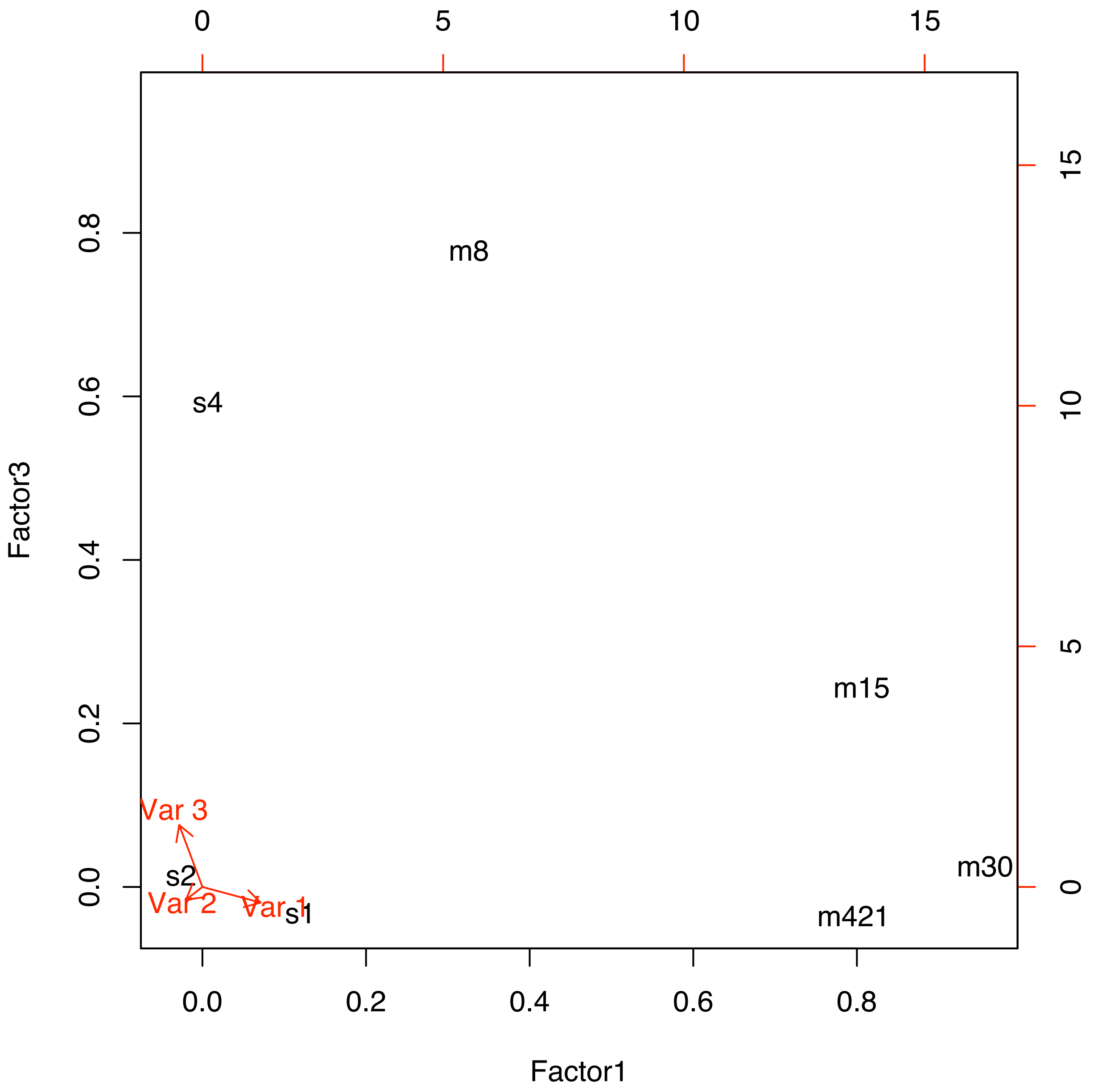
```
factanal(x = welo, factors = 3, rotation = "promax")

Uniquenesses:
   s1    s2    s4    m8   m15   m30  m421
0.076 0.005 0.088 0.005 0.037 0.018 0.192

Loadings:
     Factor1 Factor2 Factor3
s1    0.120   0.895
s2            1.006
s4            0.427   0.593
m8    0.326           0.778
m15   0.806           0.244
m30   0.957
m421  0.796   0.170

Test of the hypothesis that 3 factors are sufficient.
The chi square statistic is 4.65 on 3 degrees of freedom.
The p-value is 0.199
```

## Plotting those 3 oblique factors

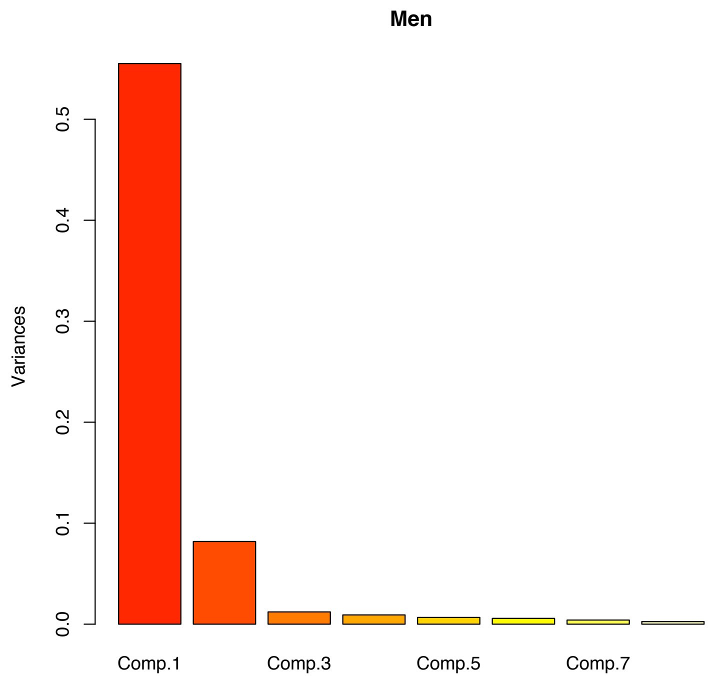


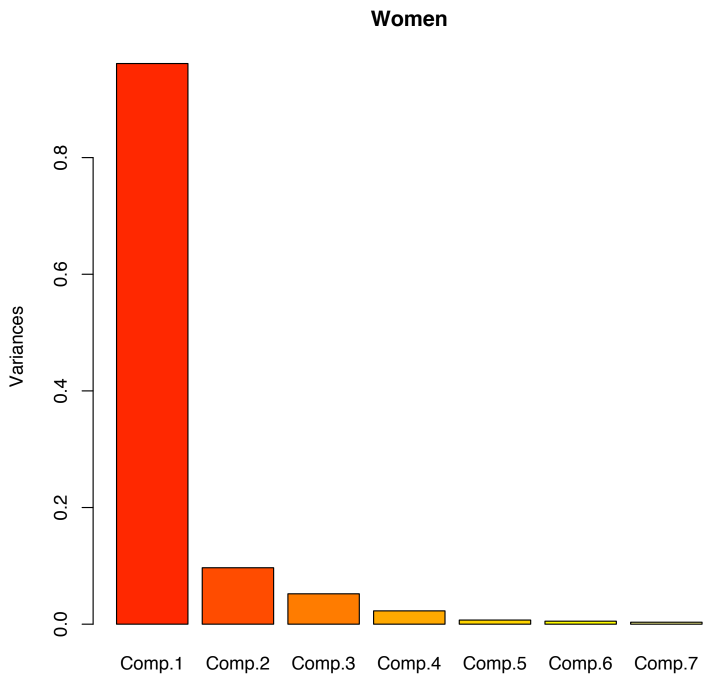
Note: p-values indicate that three factors are enough.

## Perhaps this way... but aren't there outliers?



# Not really... but there is something in women...





## But who really knows... correlation matrices

The difference between sample and fitted correlation matrix

```
2 factors
        s1   s2    s4    m8   m15   m30  m421
s1    0.16 0.00 -0.01 -0.01  0.00  0.01  0.00
s2    0.00 0.01  0.00  0.00  0.00  0.00  0.00
s4   -0.01 0.00  0.36  0.11  0.00 -0.02 -0.04
m8   -0.01 0.00  0.11  0.28  0.00 -0.02 -0.03
m15   0.00 0.00  0.00  0.00  0.03  0.00 -0.01
m30   0.01 0.00 -0.02 -0.02  0.00  0.12  0.04
m421  0.00 0.00 -0.04 -0.03 -0.01  0.04  0.46

3 factors
        s1   s2    s4   m8   m15  m30  m421
s1    0.15 0.00  0.00 0.00  0.00 0.00 -0.01
s2    0.00 0.01  0.00 0.00  0.00 0.00  0.00
s4    0.00 0.00  0.18 0.00 -0.01 0.01 -0.01
m8    0.00 0.00  0.00 0.01  0.00 0.00  0.00
m15   0.00 0.00 -0.01 0.00  0.07 0.00 -0.01
m30   0.00 0.00  0.01 0.00  0.00 0.04  0.00
m421 -0.01 0.00 -0.01 0.00 -0.01 0.00  0.38
```

## And variance-covariance matrices

The difference between sample and fitted variance matrix

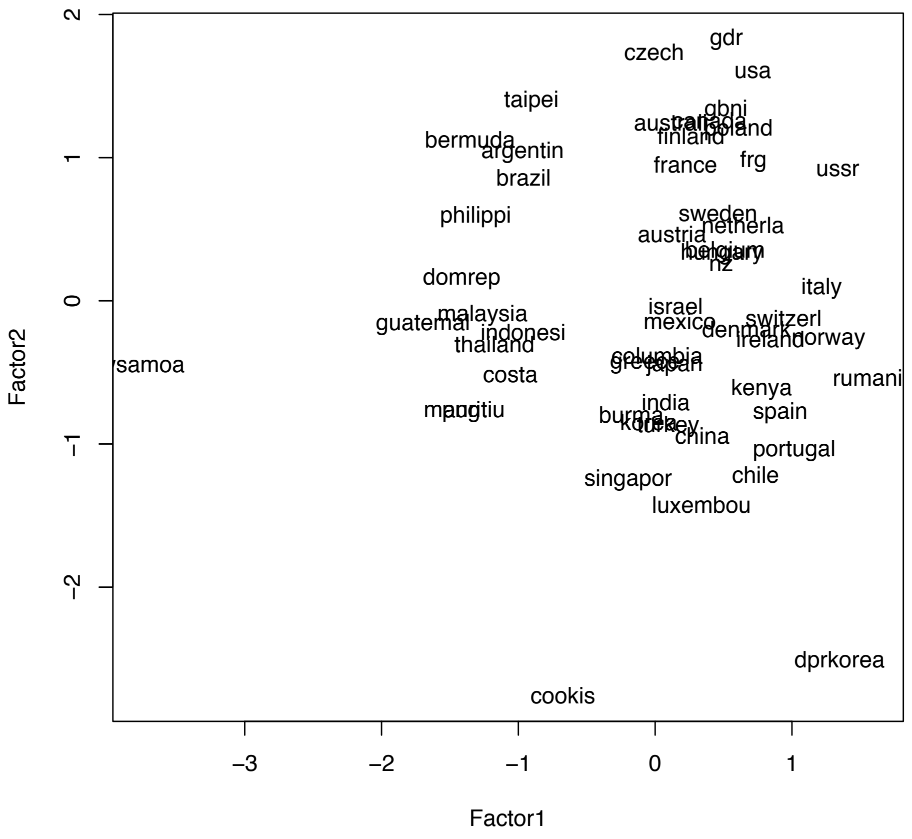
```
2 factors
         s1    s2     s4     m8   m15    m30   m421
s1   0.002 0.002  0.001  0.000 0.002  0.003  0.002
s2   0.002 0.003  0.002  0.002 0.002  0.002  0.003
s4   0.001 0.002  0.003  0.016 0.002  0.000 -0.006
m8   0.000 0.002  0.016  0.002 0.003 -0.001 -0.002
m15  0.002 0.002  0.002  0.003 0.003  0.003  0.002
m30  0.003 0.002  0.000 -0.001 0.003  0.003  0.013
m421 0.002 0.003 -0.006 -0.002 0.002  0.013  0.006

3 factors
         s1    s2    s4    m8   m15   m30  m421
s1   0.002 0.002 0.002 0.001 0.002 0.002 0.000
s2   0.002 0.003 0.002 0.002 0.002 0.002 0.003
s4   0.002 0.002 0.003 0.002 0.001 0.003 0.000
m8   0.001 0.002 0.002 0.002 0.002 0.002 0.003
m15  0.002 0.002 0.001 0.002 0.003 0.003 0.002
m30  0.002 0.002 0.003 0.002 0.003 0.003 0.004
m421 0.000 0.003 0.000 0.003 0.002 0.004 0.006
```

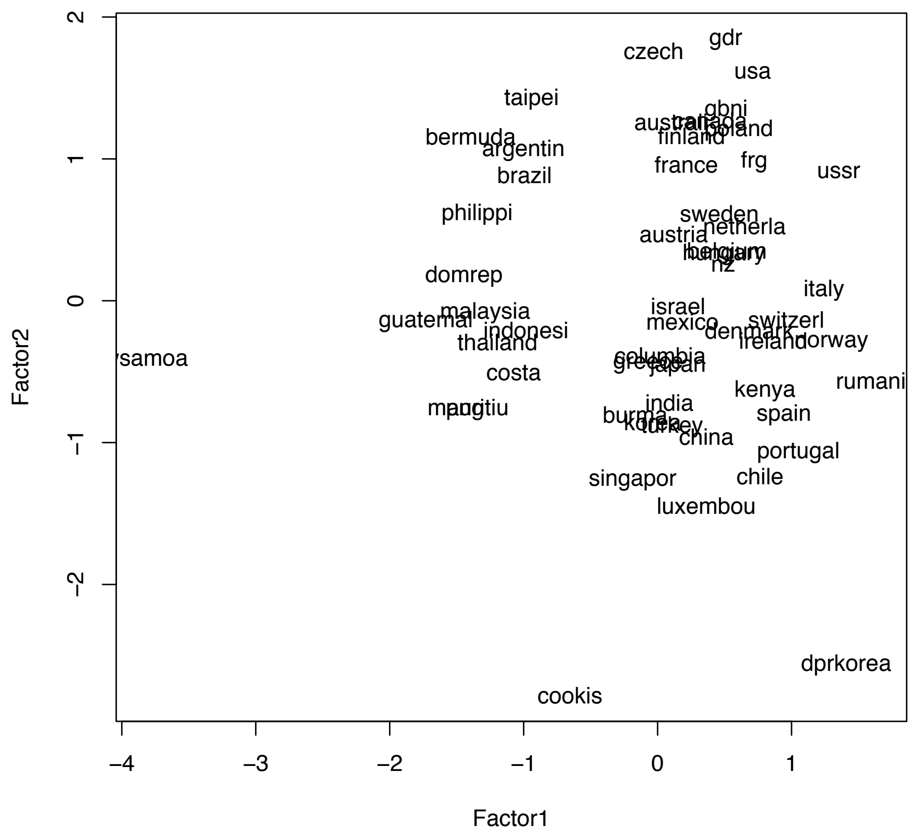
## Finally, scores

```
      Thomson         Bartlett
   Factor1 Factor2 Factor1 Factor2      Country
1  -0.9685  1.0361 -1.0018  1.0610     argentin
2   0.1143  1.2442  0.1013  1.2583     australi
3   0.1241  0.4665  0.1209  0.4708      austria
4   0.5114  0.3397  0.5180  0.3376      belgium
5  -1.3524  1.1252 -1.3950  1.1560      bermuda
6  -0.9612  0.8619 -0.9923  0.8846       brazil
7  -0.1739 -0.7955 -0.1677 -0.8032        burma
8   0.3970  1.2592  0.3898  1.2699       canada
9   0.7348 -1.2138  0.7654 -1.2380        chile
10  0.3453 -0.9417  0.3644 -0.9577        china
11  0.0148 -0.3813  0.0198 -0.3862     columbia
12 -0.6737 -2.7544 -0.6537 -2.7802       cookis
13 -1.0572 -0.5152 -1.0732 -0.5085        costa
14 -0.0072  1.7394 -0.0289  1.7611        czech
15  0.6685 -0.1988  0.6852 -0.2096      denmark
16 -1.4133  0.1522 -1.4451  0.1716       domrep
17  0.2623  1.1515  0.2535  1.1625      finland
18  0.2222  0.9521  0.2150  0.9611       france
19  0.5213  1.8263  0.5097  1.8424          gdr
20  0.7190  0.9759  0.7221  0.9791          frg
21  0.5180  1.3311  0.5124  1.3412         gbni
22 -0.0732 -0.4504 -0.0692 -0.4550       greece
23 -1.6964 -0.1641 -1.7303 -0.1451     guatemal
24  0.4889  0.3275  0.4952  0.3255      hungary
25  0.0781 -0.7086  0.0886 -0.7184        india
26 -0.9647 -0.2176 -0.9824 -0.2083     indonesi
27  0.8428 -0.2668  0.8640 -0.2806      ireland
28  0.1501 -0.0359  0.1537 -0.0382       israel
29  1.2172  0.0875  1.2419  0.0735        italy
```

```
      Thomson         Bartlett
   Factor1 Factor2 Factor1 Factor2      Country
30  0.1438 -0.4528  0.1525 -0.4602        japan
31  0.7781 -0.6194  0.8023 -0.6368        kenya
32 -0.0466 -0.8468 -0.0371 -0.8568        korea
33  1.3485 -2.5211  1.4084 -2.5692     dprkorea
34  0.3394 -1.4243  0.3643 -1.4462     luxembou
35 -1.2613 -0.1018 -1.2868 -0.0874     malaysia
36 -1.3936 -0.7575 -1.4137 -0.7496     mauritiu
37  0.1802 -0.1426  0.1858 -0.1467       mexico
38  0.6421  0.5287  0.6491  0.5272     netherla
39  0.4826  0.2514  0.4897  0.2486           nz
40  1.2676 -0.2807  1.2979 -0.2999       norway
41 -1.4162 -0.7880 -1.4364 -0.7802          png
42 -1.3113  0.5868 -1.3464  0.6104     philippi
43  0.6095  1.1938  0.6076  1.2011       poland
44  1.0166 -1.0459  1.0511 -1.0715     portugal
45  1.6000 -0.5378  1.6406 -0.5644      rumania
46 -0.1974 -1.2508 -0.1861 -1.2638     singapor
47  0.9142 -0.7825  0.9433 -0.8036        spain
48  0.4597  0.6143  0.4618  0.6162       sweden
49  0.9392 -0.1212  0.9606 -0.1343     switzerl
50 -0.9043  1.3934 -0.9407  1.4219       taipei
51 -1.1739 -0.3031 -1.1950 -0.2923     thailand
52  0.0950 -0.8768  0.1079 -0.8888       turkey
53  0.7142  1.5986  0.7094  1.6095          usa
54  1.3350  0.9125  1.3519  0.9072         ussr
55 -3.7497 -0.4584 -3.8234 -0.4175       wsamoa
```

## And their plots





## Summary: what

In factor analysis,

1. we need to (pre)determine the number of factors - either from the principal component analysis, or via testing methods

2. then estimate the loadings of the specified number of factors (nowadays mostly by the method of maximum likelihood)

3. and then we typically want to find a suitable rotation of the solution; if we want to keep common factors uncorrelated, we use only *orthogonal rotations* (like "varimax"); if we do not care about this particular aspect, we may also use *oblique rotations* (like "promax")

4. and finally, we may want to estimate the individual scores

## Summary: why

A possible objective is to arrive to some plausible interpretation of the mechanism generating the data. This was the motivation of factor analysis in its classical era - and is still vital in certain areas of application. In this context, factor analysis was frequently confused with principal component analysis, so it is important to be aware of differences: the latter is not, but the former is invariant (more precisely, equivariant) with respect to the change of scale (at least in its stochastic model, but then also in the maximum likelihood estimates)

This classical objective has been a subject of criticism: from the scientific point of view, factor interpretation has to be confirmed by other sources - otherwise, from the statistical viewpoint, factor analysis as presented here is merely a sophisticated *exploratory method*.

We can, however, also use the individual scores (which implicitly involve some dimensional reduction) for prediction. This objective received new impetus in so-called recommender systems; as such application is focused on prediction, interpretability is not an issue there

## Empiricist Relationships: Canonical Correlations

## Outline

Find directions in the first and second sample, respectively, that exhibit maximal correlation

And then second maximal, third maximal...

To summarize correlations between two samples

Alternative name: canonical variates

## Sample correlation coefficient

Recall:

$$\rho_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Properties?

## The prescription

Consider two data matrices:

$\mathbf{X}$ composed of lines $\mathbf{x}_i^\top$, and

$\mathbf{Y}$ composed of lines $\mathbf{y}_i^\top$, $i = 1, \dots, n$

Note: the numbers $p$ and $q$ of variables, the number of columns of $\mathbf{X}$ and $\mathbf{Y}$ respectively, may be different; but the number of their rows, the number of datapoints, has to be the same

We seek (nonzero) vectors $\mathbf{a}_1$ and $\mathbf{b}_1$ such that the (sample) correlation coefficient of $\mathbf{X}\mathbf{a}_1$ and $\mathbf{Y}\mathbf{b}_1$ is maximal

Once found, we may continue: seek nonzero vectors $\mathbf{a}_2$ and $\mathbf{b}_2$ such that $\mathbf{a}_2$ is orthogonal to $\mathbf{a}_1$ and $\mathbf{b}_2$ to $\mathbf{b}_1$, again maximizing correlation of $\mathbf{X}\mathbf{a}_2$ and $\mathbf{Y}\mathbf{b}_2$ now

Continuing this, we may seek nonzero vector $\mathbf{a}_j$ orthogonal to all previous $\mathbf{a}_i$, and nonzero $\mathbf{b}_j$ orthogonal to all previous $\mathbf{b}_i$ such that the correlation of $\mathbf{X}\mathbf{a}_j$ and $\mathbf{Y}\mathbf{b}_j$ is maximal. Of course, this is possible only if $j \leqslant p$ and $j \leqslant q$; hence we can repeat the above only $\min\{p, q\}$ times

## Remarks

Note that the correlation coefficient changes only by the sign of $c$ if either $\mathbf{a}_i$ or $\mathbf{b}_i$ is replaced by replaced by $c\mathbf{a}_i$ or $c\mathbf{b}_i$, respectively

Thus, we do not have to worry about negative correlations - just take the opposite $\mathbf{a}_i$ or $\mathbf{b}_i$ and they become positive. Then, when we are multiplying either $\mathbf{a}_i$ or $\mathbf{b}_i$ by positive constants, the correlations remain the same. Therefore, the only side condition on $\mathbf{a}_i$ and $\mathbf{b}_i$ is that they are nonzero - from the mathematical point of view, everything else is well-posed

From the numerical point of view, however, it may be practical to scale $\mathbf{a}_i$ and $\mathbf{b}_i$ in some convenient way; *it is usually done that the variances of resulting linear combinations of the data* - canonical variates - *are* 1

Canonical variates come in pairs, and given what was specified above, there are min$\{p, q\}$ of these pairs. In a special case when $p$ or $q$ is equal to one, there is only one pair; the maximized correlation coefficient is the **multiple correlation coefficient**

## Stochastic underpinning?

The situation here is similar as for the principal components - it is possible to consider stochastic ("population") canonical correlations (although this possibility is seldom exercised): given random vectors $\mathbf{x}$ and $\mathbf{y}$, one may seek vectors $\mathbf{a}_1$ and $\mathbf{b}_1$ maximizing the (stochastic) correlation coefficient

$$\mathrm{Cor}(\mathbf{a}_1^\top \mathbf{x}, \mathbf{b}_1^\top \mathbf{y}) = \frac{\mathrm{Cov}(\mathbf{a}_1^\top \mathbf{x}, \mathbf{b}_1^\top \mathbf{y})}{\sqrt{\mathrm{Var}(\mathbf{a}_1^\top \mathbf{x})\, \mathrm{Var}(\mathbf{b}_1^\top \mathbf{y})}}$$

and then subsequently vectors $\mathbf{a}_2, \mathbf{b}_2, \dots,$ completely analogously to the previous case

And again, in the situations when lines of $\mathbf{X}$ and $\mathbf{Y}$ are sampled respectively from the distributions of $\mathbf{x}$ and $\mathbf{y}$, the "sample" canonical correlations are empirical estimates of the "population" ones

## Solution: variance matrices again

Notation: $\mathrm{var}(\mathbf{X}) = \mathbf{S_X}$ is now denoted as $\mathbf{S_{XX}}$, and analogously $\mathrm{var}(\mathbf{Y}) = \mathbf{S_Y}$ as $\mathbf{S_{YY}}$. Symbols $\mathbf{S_{XY}} = \mathrm{cov}(\mathbf{X}, \mathbf{Y})$ and $\mathbf{S_{YX}} = \mathrm{cov}(\mathbf{Y}, \mathbf{X})$ denote now the matrices of (sample) cross-covariances between $\mathbf{X}$ and $\mathbf{Y}$, with $\mathbf{S_{XY}} = \mathbf{S_{YX}}^\top$. Then

$$\mathrm{var}\begin{pmatrix}\mathbf{X} & \mathbf{Y}\end{pmatrix} = \begin{pmatrix}\mathbf{S_{XX}} & \mathbf{S_{XY}} \\ \mathbf{S_{YX}} & \mathbf{S_{YY}}\end{pmatrix}$$

is the variance-covariance matrix of the joint data matrix $\begin{pmatrix}\mathbf{X} & \mathbf{Y}\end{pmatrix}$.

The correlation between $\mathbf{Xa}$ and $\mathbf{Yb}$ is

$$\frac{\mathbf{a}^\top \mathbf{S_{XY}} \mathbf{b}}{\sqrt{(\mathbf{a}^\top \mathbf{S_{XX}} \mathbf{a})(\mathbf{b}^\top \mathbf{S_{YY}} \mathbf{b})}}$$

## Solution continued

Then, the maximal correlation between $\mathbf{Xa}$ and $\mathbf{Yb}$,

maximized over $\mathbf{a} \neq 0$ and $\mathbf{b} \neq 0$,

is $\sqrt{\lambda}$, where $\lambda$ is the largest eigenvalue of both

$$\begin{array}{c} \mathbf{S}_{\mathbf{XX}}^{-1}\mathbf{S}_{\mathbf{XY}}\mathbf{S}_{\mathbf{YY}}^{-1}\mathbf{S}_{\mathbf{YX}} \\ \mathbf{S}_{\mathbf{YY}}^{-1}\mathbf{S}_{\mathbf{YX}}\mathbf{S}_{\mathbf{XX}}^{-1}\mathbf{S}_{\mathbf{XY}} \end{array} \text{ and } \begin{array}{c} \mathbf{a} \\ \mathbf{b} \end{array} \text{ are the corresponding eigenvectors}$$

More generally, if $\mathbf{X}$ has $p$ and $\mathbf{Y}$ has $q$ columns, then the number of canonical correlations is $m = \min\{p, q\}$ and they are the square roots of the first $m$ maximal eigenvalues of the matrices above, with coefficients being the corresponding eigenvectors

Yes, and if this $m = 1$, that is, when either $p$ or $q$ is equal to 1, then the only canonical correlation is called **multiple correlation coefficient**, of one variable against the reminding ones

In stochastic version, the solution proceeds in the completely analogous way, taking stochastic counterparts of the variances and covariances

## Example: sons

Length and breadth, respectively, of the head of the first and second son

| | L1 | B1 | L2 | B2 |
|---|---|---|---|---|
| 1 | 191 | 155 | 179 | 145 |
| 2 | 195 | 149 | 201 | 152 |
| 3 | 181 | 148 | 185 | 149 |
| 4 | 183 | 153 | 188 | 149 |
| 5 | 176 | 144 | 171 | 142 |
| 6 | 208 | 157 | 192 | 152 |
| 7 | 189 | 150 | 190 | 149 |
| 8 | 197 | 159 | 189 | 152 |
| 9 | 188 | 152 | 197 | 159 |
| 10 | 192 | 150 | 187 | 151 |
| 11 | 179 | 158 | 186 | 148 |
| 12 | 183 | 147 | 174 | 147 |
| 13 | 174 | 150 | 185 | 152 |
| 14 | 190 | 159 | 195 | 157 |
| 15 | 188 | 151 | 187 | 158 |
| 16 | 163 | 137 | 161 | 130 |
| 17 | 195 | 155 | 183 | 158 |
| 18 | 186 | 153 | 173 | 148 |
| 19 | 181 | 145 | 182 | 146 |
| 20 | 175 | 140 | 165 | 137 |
| 21 | 192 | 154 | 185 | 152 |
| 22 | 174 | 143 | 178 | 147 |
| 23 | 176 | 139 | 176 | 143 |
| 24 | 197 | 167 | 200 | 158 |
| 25 | 190 | 163 | 187 | 150 |

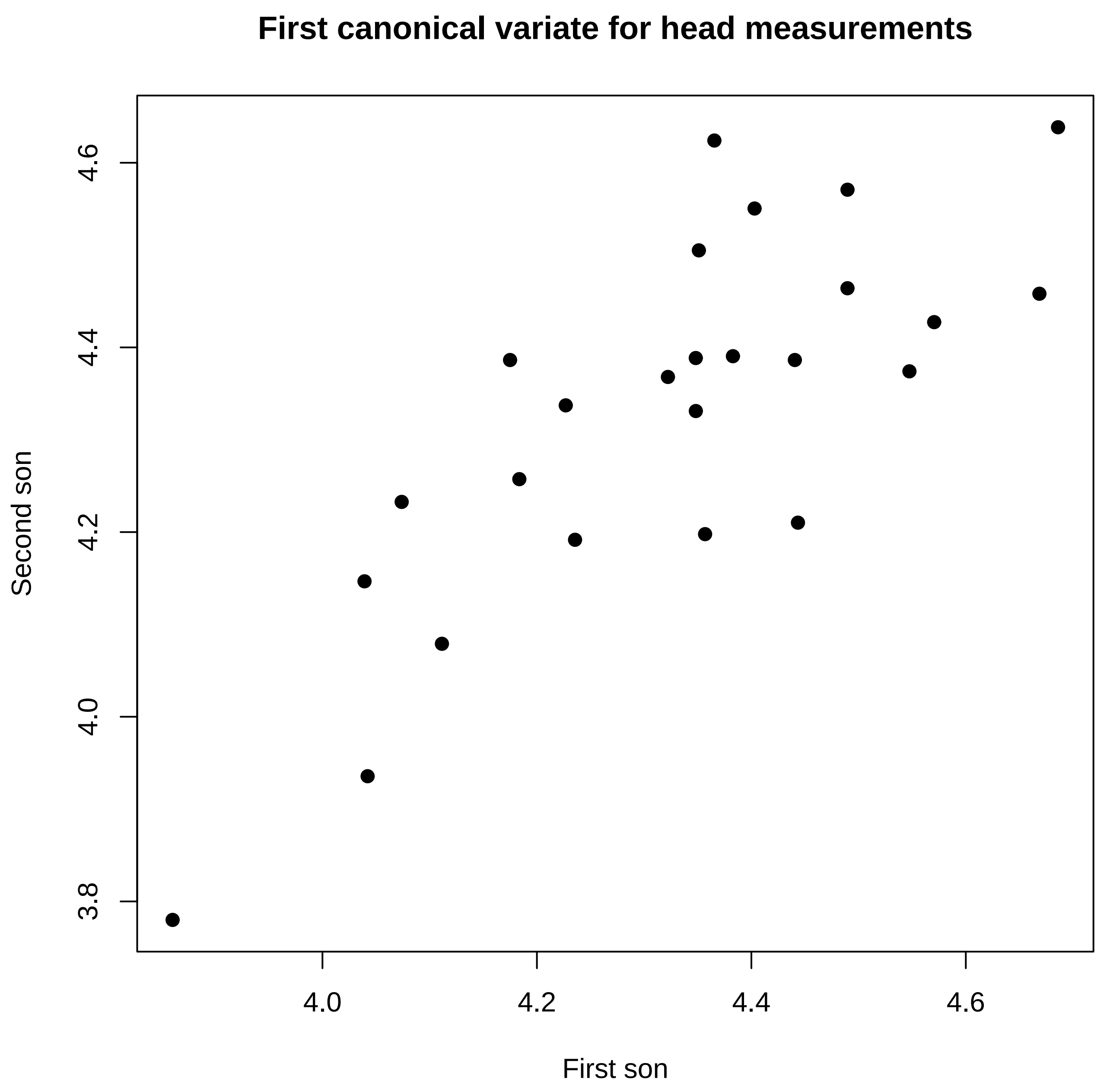
## The result

```
> sons <- read.table("sons.d")
> cancor(sons[,1:2],sons[,3:4])
$cor
[1] 0.7885079 0.0537397
$xcoef
         [,1]        [,2]
L1 0.01154653 -0.02857148
B1 0.01443910  0.03816093
$ycoef
         [,1]        [,2]
L2 0.01025573 -0.03595605
B2 0.01637533  0.05349758
$xcenter
    L1     B1
185.72 151.12
$ycenter
    L2     B2
183.84 149.24
```

## Canonical variates?

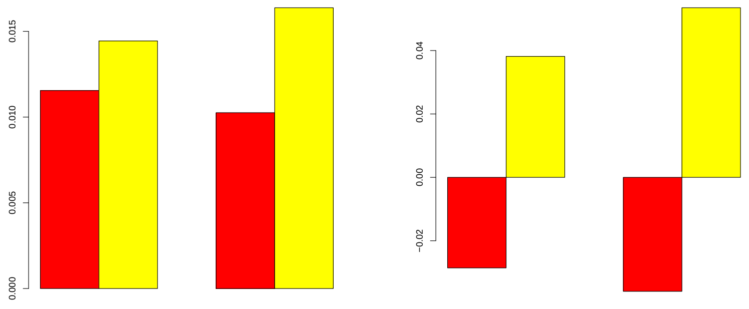
```
> sons.cc <- cancor(sons[,1:2],sons[,3:4])
> canvarx <- as.matrix(sons[,1:2]) %*% sons.cc$xcoef[,1]
> canvary <- as.matrix(sons[,3:4]) %*% sons.cc$ycoef[,1]
> plot(canvarx, canvary, pch=16,
+ xlab='First son',ylab='Second son')
```

# And the plot

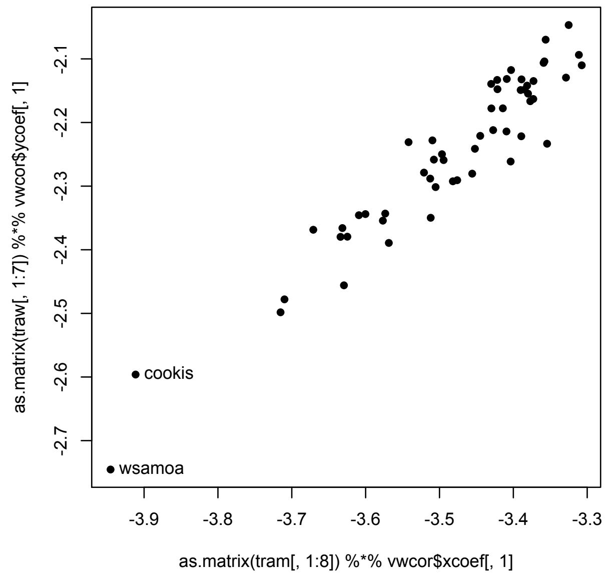

First canonical variate for head measurements
Second son
First son
3.8
4.0
4.2
4.4
4.6
4.0
4.2
4.4
4.6

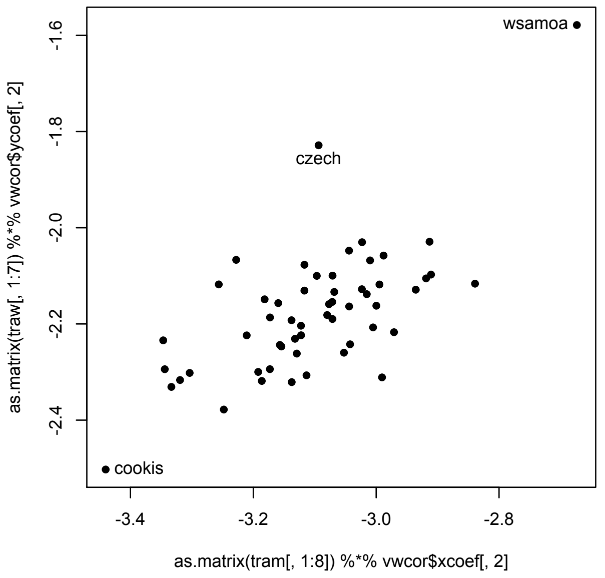
## What else?

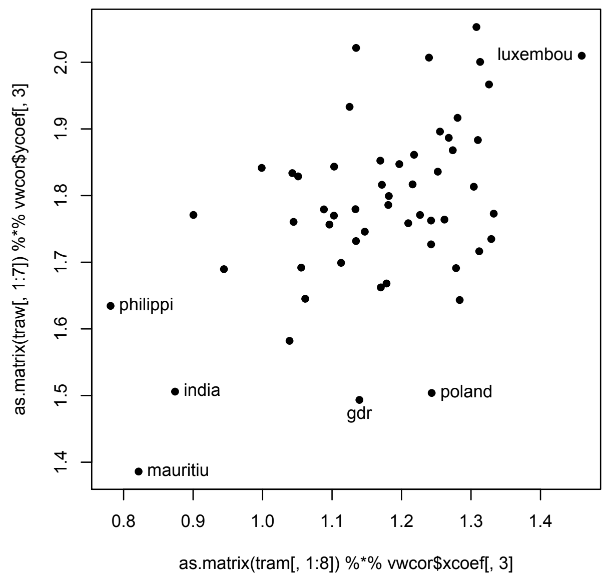
```
> var(sons)
         L1       B1        L2       B2
L1 95.29333 52.86833  69.66167 46.11167
B1 52.86833 54.36000  51.31167 35.05333
L2 69.66167 51.31167 100.80667 56.54000
B2 46.11167 35.05333  56.54000 45.02333

$xcoef
         [,1]        [,2]
L1 0.01154653 -0.02857148
B1 0.01443910  0.03816093

$ycoef
         [,1]        [,2]
L2 0.01025573 -0.03595605
B2 0.01637533  0.05349758
```

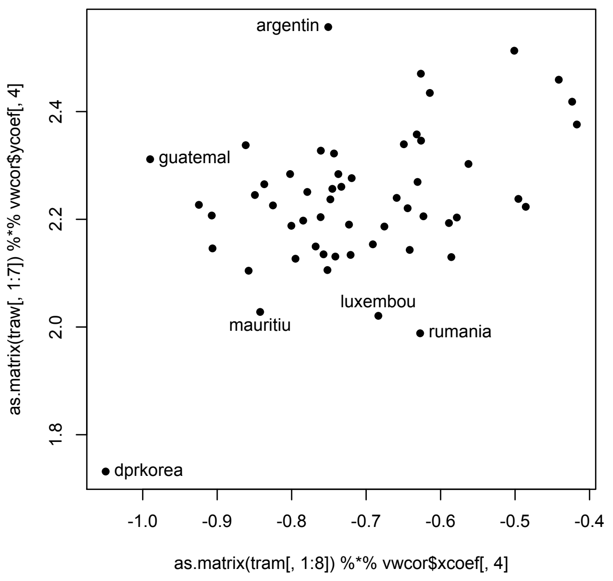


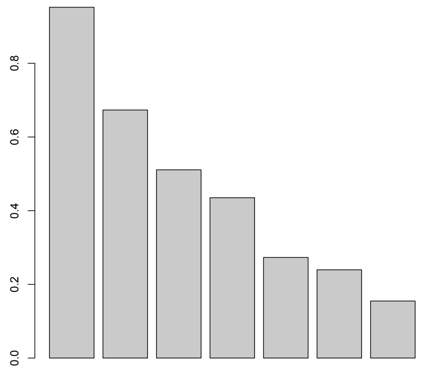
# Example: running data









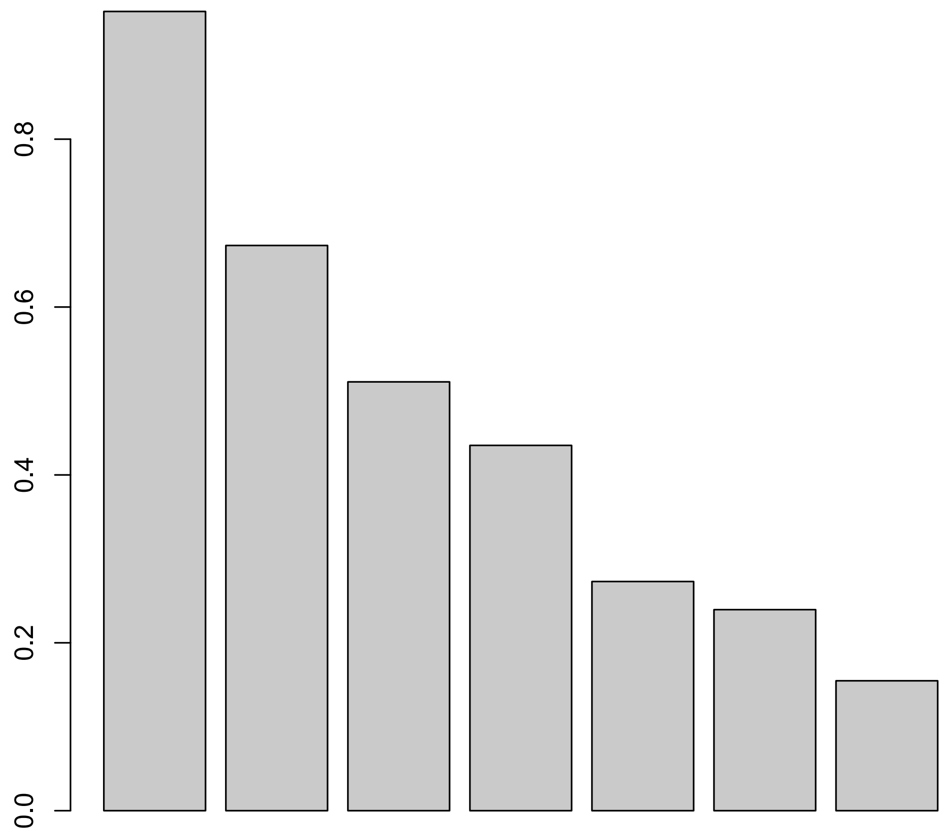


## Running data transcript

```
> vwcor=cancor(tram[,1:8],traw[,1:7])
> par(mfrow=c(2,2))
> plot(as.matrix(tram[,1:8]) %*% vwcor$xcoef[,1],
+ as.matrix(traw[,1:7]) %*% vwcor$ycoef[,1],pch=16)
> identify(as.matrix(tram[,1:8]) %*% vwcor$xcoef[,1],
+ as.matrix(traw[,1:7]) %*% vwcor$ycoef[,1],labels=traw[,8])
[1] 12 55
> plot(as.matrix(tram[,1:8]) %*% vwcor$xcoef[,2],
...
[1] 12 14 55
> plot(as.matrix(tram[,1:8]) %*% vwcor$xcoef[,3],
...
[1] 19 25 34 36 42 43
> plot(as.matrix(tram[,1:8]) %*% vwcor$xcoef[,4],
...
[1]  1 23 33 34 36 45
```

## How canonical correlations fare here?

```
> barplot{vwcor}
```



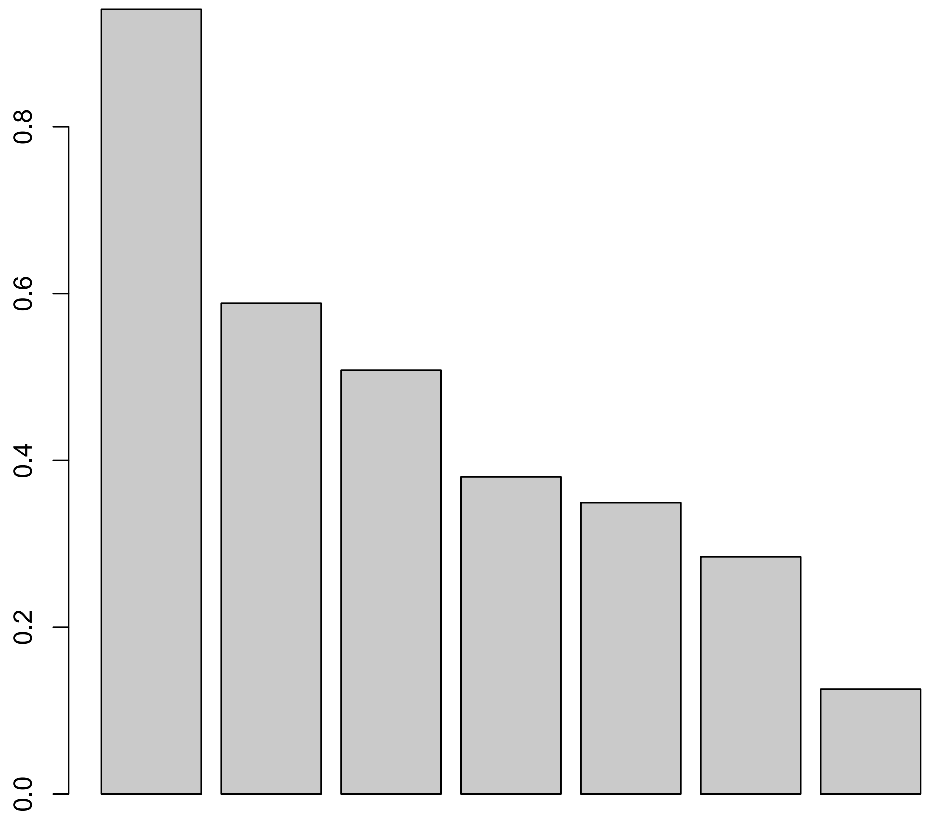
## Running data transcript continued

```
> vwcor
$cor
[1] 0.9521159 0.6733411 0.5108601 0.4351771 0.2730063 0.2395034 0.1547629
$xcoef
             [,1]         [,2]         [,3]        [,4]          [,5]         [,6]
s1    0.104094168 -0.107380162 -0.515664832 -0.25908559 -0.6883280074  0.376619624
s2   -0.108737342 -0.106936978 -0.109330683  0.06559543  0.4299810105 -0.217250889
s4    0.013809246 -0.124909061  0.174467768  0.08240273  0.0093848913  0.041841603
m8   -0.846947526  2.292211283  2.128900905 -4.67774851 -0.5470794211 -0.441871257
m15  -0.206532529  0.277130648 -0.415851876  1.31836018 -0.0001939217 -1.524535704
m50  -0.008423944 -0.080300588 -0.030690671  0.01810175 -0.5673227355 -0.309327155
m100 -0.024762400  0.087875854 -0.007190537 -0.10191559  0.2682108659  0.241688938
m421  0.001308979 -0.003716701 -0.006534732  0.02209632 -0.0084966587  0.006218362
...
$ycoef
              [,1]          [,2]         [,3]          [,4]         [,5]        [,6]
s1   -0.0130055307  0.0176386680  0.061917582  0.5792507543  0.216963524 -0.64954132
s2   -0.0605052422 -0.0957820269 -0.015752815 -0.3076449771  0.097987108  0.13392170
s4    0.0209834118 -0.0064488138  0.073967723 -0.0029267579 -0.112539112  0.04433569
m8   -0.5073336300 -1.6614889659 -2.314920945  2.2016538222  1.758300290  0.07128654
m15   0.0094233762  1.0338571845  1.374913004 -0.2205103685  0.701931326  0.69584361
m30  -0.0694585560 -0.0872183745 -0.281419643 -0.0572468827 -0.469592348 -0.36565080
m421 -0.0007024018  0.0001743919 -0.005775172 -0.0008261367  0.001848462  0.00326083
...
```

## How about omitting very small countries?

(Cook Island, Luxembourg, Western Samoa omitted)

```
> vwnew=cancor(tram[-c(12,34,55),1:8],traw[-c(12,34,55),1:7])
> barplot(vwnew$cor)
```



## The result

```
> vwnew
$cor
[1] 0.9407974 0.5884286 0.5081459 0.3802782 0.3493206 0.2844400 0.1257613

$xcoef
              [,1]        [,2]         [,3]        [,4]        [,5]          [,6]
s1   -0.0592844998  0.42333578 -0.092313052  0.36266317  1.21069000  0.2341572681
s2   -0.0586454028 -0.23234027 -0.316222887 -0.32816777 -0.39066849 -0.1487345454
s4    0.0081157972 -0.10537898  0.213525556 -0.03270155  0.06395563 -0.0241539365
m8   -0.6341725899 -0.58367704 -0.768746039  4.99394224 -2.81142541  0.0256325079
m15  -0.2088342032 -0.06469008  0.085591407 -1.70388403 -0.50357199  0.4721261302
m50  -0.0603341980 -0.01619295 -0.031304150  0.14973689  0.15662788  0.7664245903
m100 -0.0225362139  0.05796283 -0.021212688  0.14387843 -0.07560743 -0.3486315644
m421  0.0007541957  0.01502310  0.006513154 -0.02803850  0.01260490  0.0005577576
...
$ycoef
              [,1]         [,2]         [,3]         [,4]         [,5]         [,6]
s1   -0.0003540832  0.050271374  0.178529200 -0.817049127 -0.375775582  0.077606779
s2   -0.0593500609 -0.203584049 -0.148173170  0.308869741 -0.006811645 -0.139683776
s4    0.0158728665  0.009585147  0.060670314  0.040390516  0.097630689  0.094526166
m8   -0.3105236092 -1.532243337 -0.262419707 -3.163682685 -0.185052036 -2.292433786
m15  -0.1714313947  1.075555404  1.094736369  1.344590415 -0.446258250 -0.552282755
m30  -0.0500201340 -0.116154880 -0.351132266 -0.093779920 -0.098462780  0.545947847
m421 -0.0011833621  0.003367182 -0.004573076 -0.002911054  0.005610899 -0.004093673
...
```

## Canonical correlations via SVD

We start again form the variance-covariance matrix of joint data matrix $(\mathbf{X}\ \mathbf{Y})$ and multiply it by matrices from left and right as follows:

$$\begin{pmatrix} \mathbf{S}_{\mathbf{XX}}^{-1/2} & \mathbf{O} \\ \mathbf{O} & \mathbf{S}_{\mathbf{YY}}^{-1/2} \end{pmatrix} \begin{pmatrix} \mathbf{S}_{\mathbf{XX}} & \mathbf{S}_{\mathbf{XY}} \\ \mathbf{S}_{\mathbf{YX}} & \mathbf{S}_{\mathbf{YY}} \end{pmatrix} \begin{pmatrix} \mathbf{S}_{\mathbf{XX}}^{-1/2} & \mathbf{O} \\ \mathbf{O} & \mathbf{S}_{\mathbf{YY}}^{-1/2} \end{pmatrix}$$
$$= \begin{pmatrix} \mathbf{I} & \mathbf{S}_{\mathbf{XX}}^{-1/2}\mathbf{S}_{\mathbf{XY}}\mathbf{S}_{\mathbf{YY}}^{-1/2} \\ \mathbf{S}_{\mathbf{YY}}^{-1/2}\mathbf{S}_{\mathbf{YX}}\mathbf{S}_{\mathbf{XX}}^{-1/2} & \mathbf{I} \end{pmatrix}$$

where $\mathbf{A}^{-1/2}$ a matrix such that $\mathbf{A}^{-1/2}\mathbf{A}^{-1/2} = \mathbf{A}^{-1}$

We then apply SVD to the matrix (the other one is its transpose)

$$\mathbf{S}_{\mathbf{XX}}^{-1/2}\mathbf{S}_{\mathbf{XY}}\mathbf{S}_{\mathbf{YY}}^{-1/2} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{V}^{\top}$$

The singular values are canonical correlations now; the columns of matrices $\mathbf{S}_{\mathbf{XX}}^{-1/2}\mathbf{U}$ and $\mathbf{S}_{\mathbf{YY}}^{-1/2}\mathbf{V}$ give respectively the vectors of coefficients.

In the stochastic version, we again proceed completely analogously, using the stochastic counterparts of variances and covariances

## It works for categorical data too

We can apply the technology to the qualitative/categorical data with two variables. Recall hair/eye color/sex data: we form the aggregated hair/eye data

```
> haireye=apply(HairEyeColor,1:2,sum)
> haireye
       Eye
Hair    Brown Blue Hazel Green
  Black    68   20    15     5
  Brown   119   84    54    29
  Red      26   17    14    14
  Blond     7   94    10    16
```

## R shenanigans

Next step is to convert the tabular data to the "data frame" form

```
> hairdat=haireye[rep(row.names(haireye),haireye$Freq),1:2]
Error in haireye$Freq : $ operator is invalid for atomic vectors
> hair=as.table(haireye)
> hairdat=hair[rep(row.names(hair),hair$Freq),1:2]
Error in hair$Freq : $ operator is invalid for atomic vectors
> hair=as.data.frame(haireye)
> hairdat=hair[rep(row.names(hair),hair$Freq),1:2]
Error in rep(row.names(hair), hair$Freq) : invalid 'times' argument
> hair=as.data.frame(as.table(haireye))
> hairdat=hair[rep(row.names(hair),hair$Freq),1:2]
```

Finally!! And then we form the "dummy" 0-1 variables out of those: as there are 4 levels of each variable ("factor"), there are 4 variables in each of the following matrices: each variable is equal to 1 when the particular level of the factor (color) is present at the $i$-th object, and to 0 otherwise

```
> X=model.matrix(~Hair-1,data=hairdat)
> Y=model.matrix(~Eye-1,data=hairdat)
```

## And now: hocus-pocus

```
> cancor(X,Y)
$cor
[1] 0.45691646 0.14908593 0.05097489
...
```

A technique built on this is called **correspondence analysis**

```
> corresp(haireye,nf=3)
First canonical correlation(s): 0.45691646 0.14908593 0.05097489
...
```

Some quantities which will be needed later:

```
> r = apply(haireye,1,sum)/sum(haireye)
> c = apply(haireye,2,sum)/sum(haireye)
> E = haireye/sum(haireye)
```

## Correspondence analysis

Suppose that $\mathbf{E}$ is the matrix formed from $n_{ij}/n$ (the estimates of cell probabilities not assuming the independence hypothesis) and $\mathbf{R}$, $\mathbf{C}$ are diagonal matrices formed from vectors $\mathbf{r}$ and $\mathbf{c}$, with elements $r_i = n_{i\cdot}/n$ and $c_j = n_{\cdot j}/n$, respectively.

Consider the matrix $\mathbf{R}^{-1/2}\mathbf{E}\mathbf{C}^{-1/2}$ (it is easy to form the square roots of diagonal matrices with positive elements). This matrix has elements

$$\frac{e_{ij}}{\sqrt{r_i c_j}}$$

SVD of this matrix can be viewed as returning scores giving maximal "correlations" for rows and columns. The largest singular value is always one, corresponding to constant scores; hence we dismiss it, and look only for nontrivial solutions corresponding to singular values beginning with the second largest. That is, we form the SVD of

$$\mathbf{R}^{-1/2}(\mathbf{E} - rc^{\top})\mathbf{C}^{-1/2} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{V}^{\top}$$

instead, and then we may take first one or two singular values.

```
> svd(diag(1/sqrt(r)) %*% E %*% diag(1/sqrt(c)))
$d
[1] 1.00000000 0.45691646 0.14908593 0.05097489
$u
           [,1]        [,2]       [,3]       [,4]
[1,] -0.4271211  0.47166009  0.6154461  0.4651134
[2,] -0.6950598  0.22552151 -0.1522951 -0.6654608
[3,] -0.3463126  0.09817011 -0.7424993  0.5649115
[4,] -0.4631706 -0.84678181  0.2161646  0.1473309
$v
           [,1]       [,2]       [,3]        [,4]
[1,] -0.6096078  0.6566258  0.3611439  0.25844921
[2,] -0.6026406 -0.7220003  0.3353208 -0.05567605
[3,] -0.3963516  0.1844169 -0.4450167 -0.78157274
[4,] -0.3287980 -0.1163980 -0.7477267  0.56502056
> svd(diag(1/sqrt(r)) %*% (E - r %*% t(c)) %*% diag(1/sqrt(c)))
$d
[1] 4.569165e-01 1.490859e-01 5.097489e-02 2.929785e-19
$u
            [,1]       [,2]       [,3]      [,4]
[1,] -0.47166009  0.6154461 -0.4651134 0.4271211
[2,] -0.22552151 -0.1522951  0.6654608 0.6950598
[3,] -0.09817011 -0.7424993 -0.5649115 0.3463126
[4,]  0.84678181  0.2161646 -0.1473309 0.4631706
$v
           [,1]       [,2]        [,3]       [,4]
[1,] -0.6566258  0.3611439 -0.25844921 -0.6096078
[2,]  0.7220003  0.3353208  0.05567605 -0.6026406
[3,] -0.1844169 -0.4450167  0.78157274 -0.3963516
[4,]  0.1163980 -0.7477267 -0.56502056 -0.3287980
```

## The mechanized way

```
> library(MASS)
> hc1 = corresp(haireye)
First canonical correlation(s): 0.4569165
 Hair scores:
     Black      Brown        Red      Blond
-1.1042772 -0.3244635 -0.2834725  1.8282287
 Eye scores:
     Brown       Blue      Hazel      Green
-1.0771283  1.1980612 -0.4652862  0.3540108

> hc2 = corresp(haireye,nf=2)
First canonical correlation(s): 0.4569165 0.1490859
 Hair scores:
            [,1]       [,2]
Black -1.1042772  1.4409170
Brown -0.3244635 -0.2191109
Red   -0.2834725 -2.1440145
Blond  1.8282287  0.4667063
 Eye scores:
            [,1]       [,2]
Brown -1.0771283  0.5924202
Blue   1.1980612  0.5564193
Hazel -0.4652862 -1.1227826
Green  0.3540108 -2.2741218
```

## Interpretation I

Can be thought of as analysis of the $\chi^2$ statistic for independence, because the elements of the matrix $\mathbf{R}^{-1/2}(\mathbf{E}-\mathbf{rc}^\top)\mathbf{C}^{-1/2}$ are Pearson residuals, up to a factor $\sqrt{n}$.

```
> class(haireye)='table'
> matrix(residuals(glm(Freq~Hair+Eye,family=poisson,
data=as.data.frame(haireye)),type='pearson'),4,4)/sqrt(sum(haireye))
             [,1]        [,2]        [,3]        [,4]
[1,]  0.180773066 -0.12615064 -0.01961905 -0.08029590
[2,]  0.050694815 -0.08012300  0.05561963 -0.01418351
[3,] -0.003081574 -0.07110772  0.03502737  0.09381990
[4,] -0.240474512  0.28973637 -0.09156384  0.02518174
> diag(1/sqrt(r)) %*% (E - r %*% t(c)) %*% diag(1/sqrt(c))
             [,1]        [,2]        [,3]        [,4]
[1,]  0.180773066 -0.12615064 -0.01961905 -0.08029590
[2,]  0.050694815 -0.08012300  0.05561963 -0.01418351
[3,] -0.003081574 -0.07110772  0.03502737  0.09381990
[4,] -0.240474512  0.28973637 -0.09156384  0.02518174
```

## Recall: the anatomy of $\chi^2$-statistic

Recall: we want to test independence in a $r \times c$ contingency table. The probabilities of observations in cells are $p_{ij}$; under the hypothesis of independence, $p_{ij} = p_{i\cdot}p_{\cdot j}$, where $p_{i\cdot}$ and $p_{\cdot j}$ are column and row sums - marginal probabilities. We observe cells frequencies $n_{ij}$; the estimates for $p_{i\cdot}$, and $p_{\cdot j}$ are $\widehat{p}_{i\cdot} = n_{i\cdot}/n$ and $\widehat{p}_{\cdot j} = n_{\cdot j}/n$, respectively; $n$ is the total number of observations. Under the hypothesis of independence, the estimate for the cell probability is $\widehat{p}_{ij} = \widehat{p}_{i\cdot}\widehat{p}_{\cdot j} = (n_{i\cdot}/n)(n_{\cdot j}/n)$, and therefore, while the observed number is $O = n_{ij}$, the predicted number of observations is

$$P = n\,\frac{n_{i\cdot}n_{\cdot j}}{n^2} = \frac{n_{i\cdot}n_{\cdot j}}{n}; \qquad \text{the test statistics is} \qquad \sum_{\text{all cells}} \frac{(O_k - P_k)^2}{P_k}$$

- the sum of squares of "Pearson residuals" $\dfrac{O_k - P_k}{\sqrt{P_k}}$

We use $\chi^2$ distribution with $(r-1)(c-1)$ degrees of freedom to assess how large is this statistic is large enough, via its right tail value, which gives the p-value for the hypothesis of independence. If this p-value is low, we may reject the hypothesis: good, but what then?

## Another interpretation - and plotting

Correspondence analysis can be also viewed, as mentioned above, as canonical correlations : a search for the linear combination giving maximal contingency ("correlation") - not accounting for the trivial constant solution

In this direction, it can be a comparison of distances between "profiles" - rowwise or columnwise conditional distributions.

If the resulting SVD is $\mathbf{U}\boldsymbol{\Lambda}\mathbf{V}^\top$, then of interest are first columns of $\mathbf{A} = \mathbf{R}^{-1/2}\mathbf{U}\boldsymbol{\Lambda}$ and $\mathbf{B} = \mathbf{C}^{-1/2}\mathbf{V}\boldsymbol{\Lambda}$

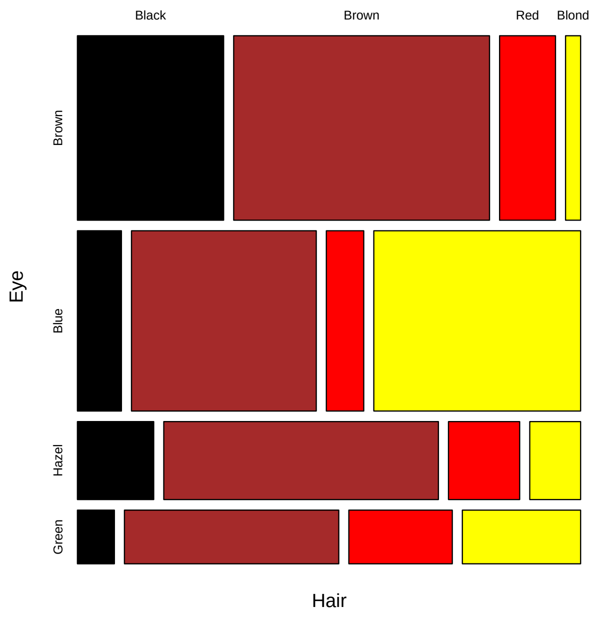
Inertia: the sum of squares of omitted singular values

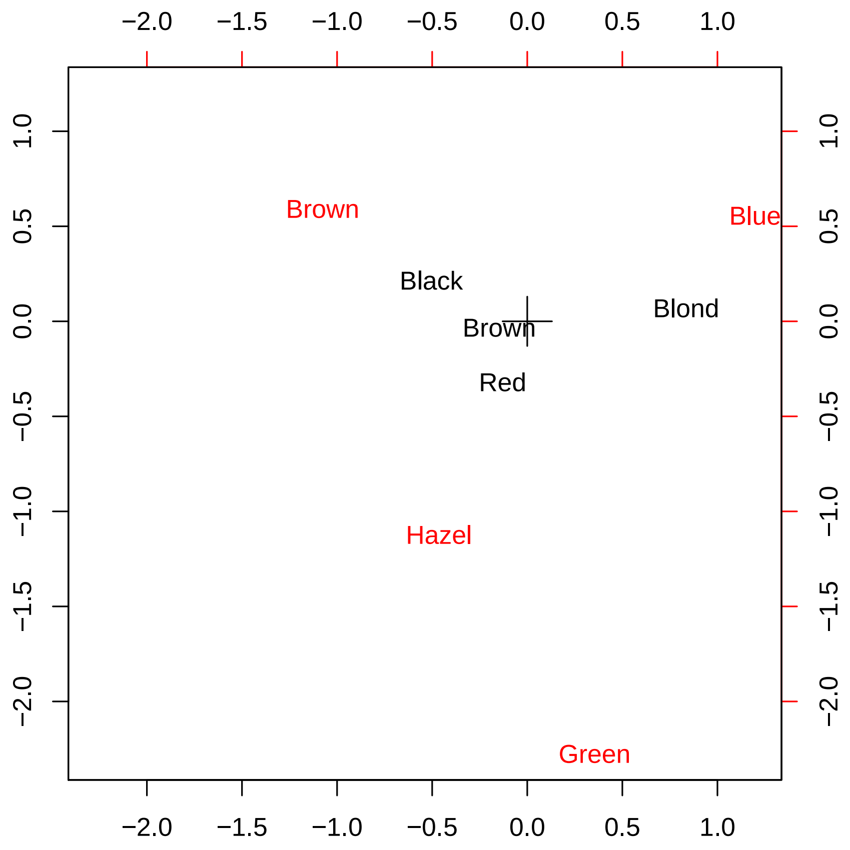
Plotting:

"Asymmetric approach": plots either first two columns of $\mathbf{A}$ with first two columns of $\mathbf{C}^{-1/2}\mathbf{V}$ (rows) or $\mathbf{B}$ with first two columns of $\mathbf{R}^{-1/2}\mathbf{U}$ (columns). Row plot can be viewed as that $\mathbf{A}$ is a convex combination of row profiles, given by $\mathbf{C}^{-1/2}\mathbf{V}$ - and similarly the column one

Or, in "classical correspondence analysis", "symmetric approach", first two columns of $\mathbf{A}$ and $\mathbf{B}$ are plotted on the same figure

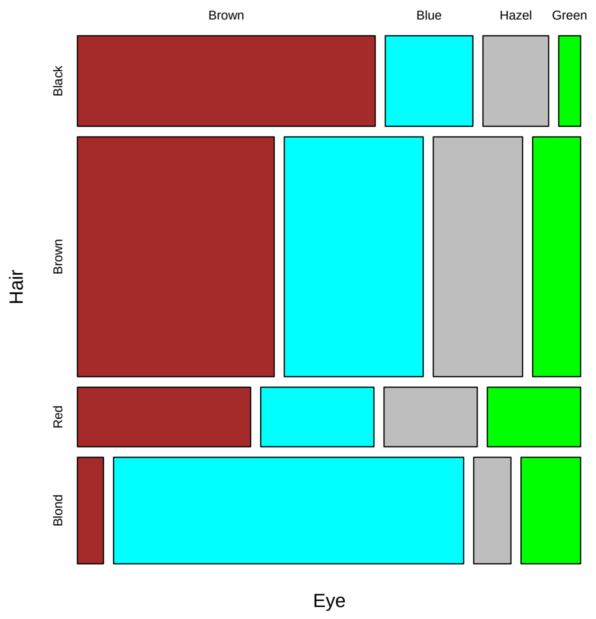
# Asymmetric view: rows

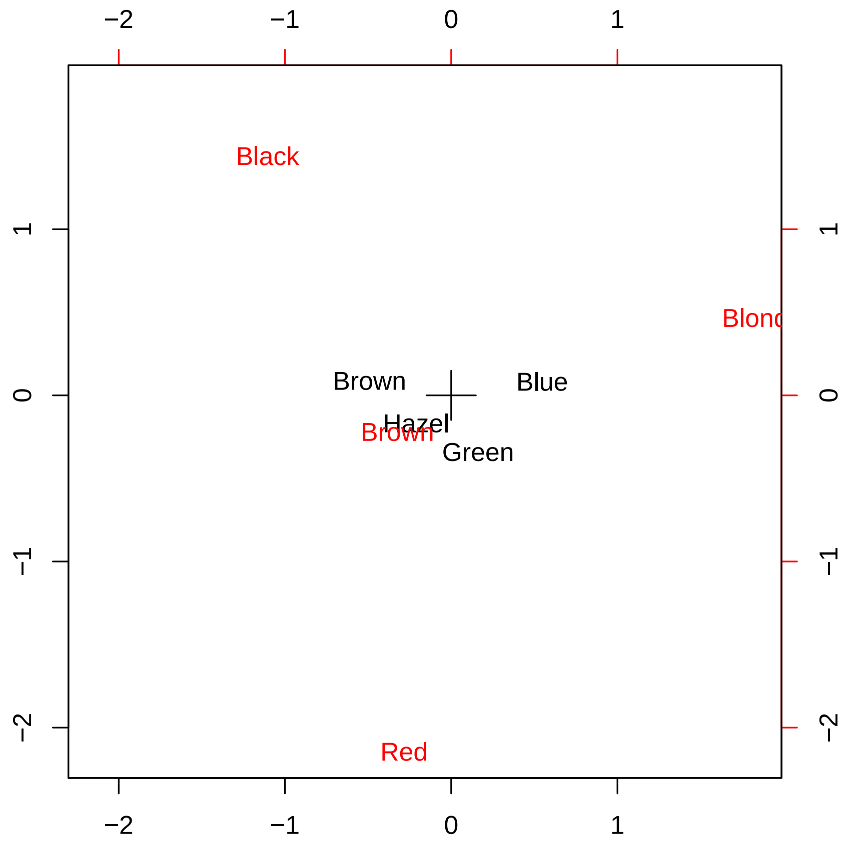




```
> plot(hc2,type="rows")
> biplot(hc2,type="rows")
> mosaicplot(haireye,sort=c(2,1),dir=c('v','h'))
```

# Asymmetric view: columns

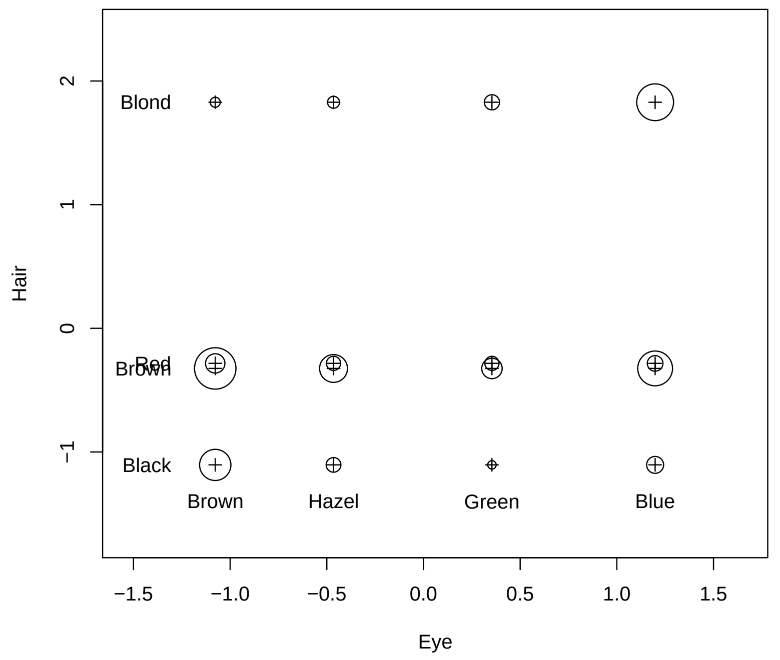




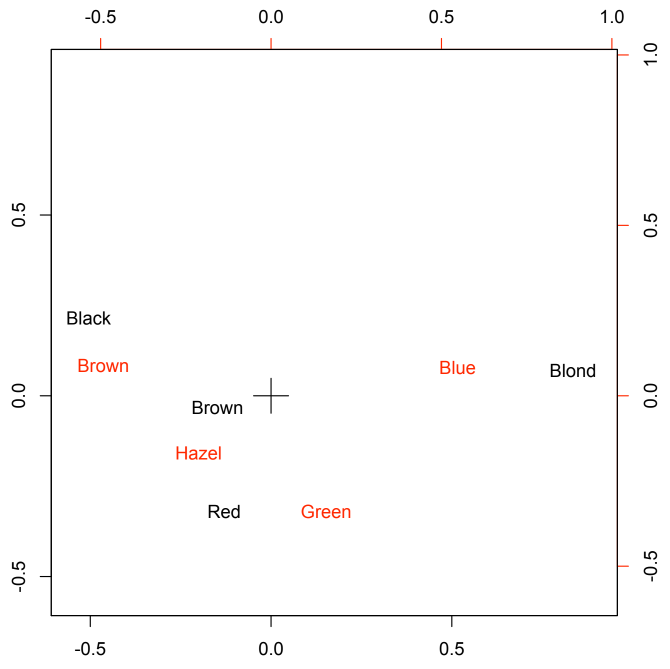
```
> plot(hc2,type="columns")
> biplot(hc2,type="columns")
> mosaicplot(haireye,sort=c(1,2),dir=c('h','v'))
```

# And now, symmetric view

First column only



First two columns (more common)



```
> plot(hc2)
> biplot(hc2)
> biplot(hc2,xlim=c(-0.55,0.9),ylim=c(-0.55,0.9))
```

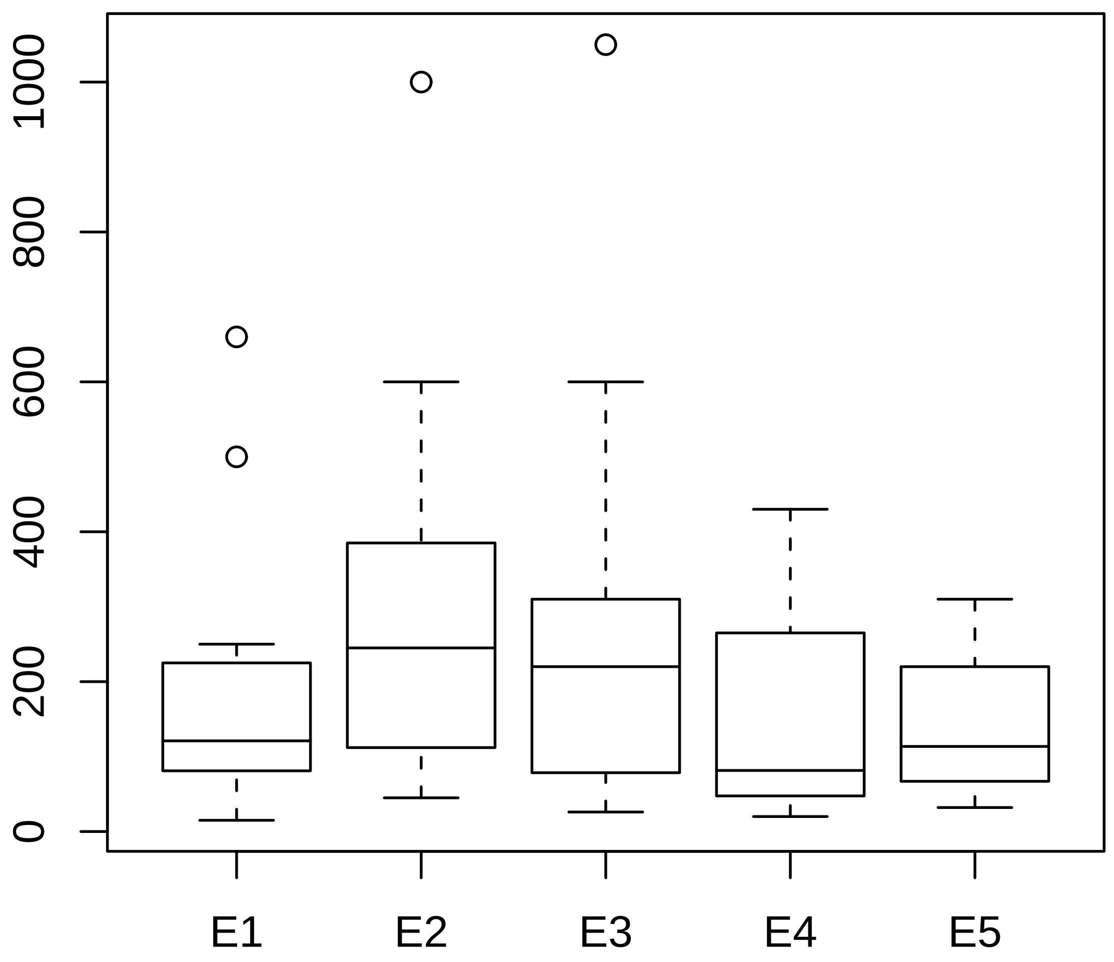
## Extending Classics: Multivariate Normal Distribution and All Around

## Recall: electrodes

Various electrodes have been tried successively on arms of 16 subjects.

```
> read.table("electro.d",header=T,row.names=1)
    E1   E2   E3  E4  E5
1  500  400   98 200 250
2  660  600  600  75 310
3  250  370  220 250 220
4   72  140  240  33  54
5  135  300  450 430  70
6   27   84  135 190 180
7  100   50   82  73  78
8  105  180   32  58  32
9   90  180  220  34  64
10 200  290  320 280 135
11  15   45   75  88  80
12 160  200  300 300 220
13 250  400   50  50  92
14 170  310  230  20 150
15  66 1000 1050 280 220
16 107   48   26  45  51
```

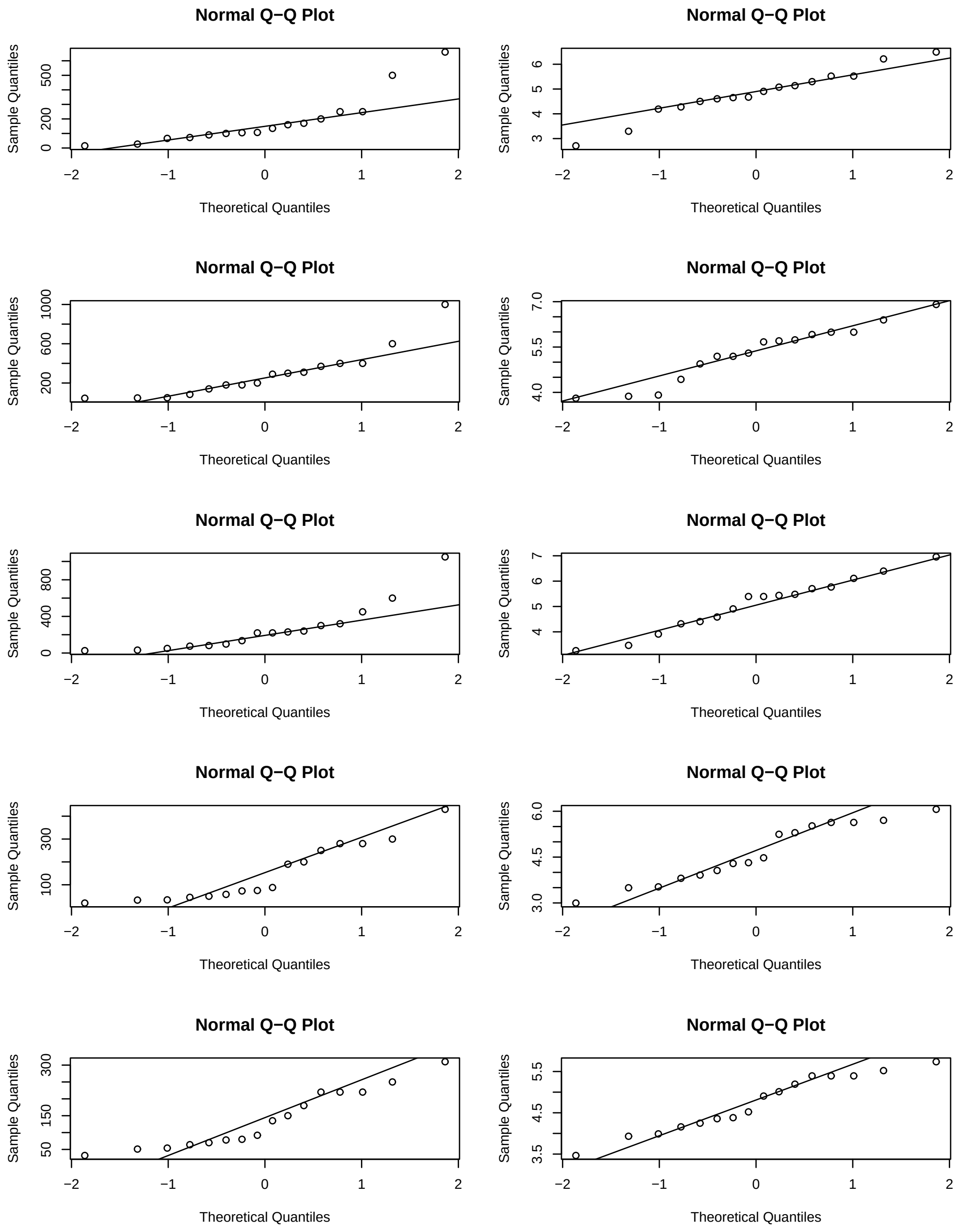
## A picture (to figure out whether normal distribution of results is plausible here)



## Seems like a bit of asymmetry there... take logs!

```
> elel=log(read.table("electro.d",header=T,row.names=1))
> par(mfcol=c(5,2))
> for (i in 1:5)
+ {
+ qqnorm(ele[,i])
+ qqline(ele[,i])
+ }
> for (i in 1:5)
+ {
+ qqnorm(elel[,i])
+ qqline(elel[,i])
+ }
```

# Normal qqplots: left before, right after the logs



## Normal distribution: definition

The most straightforward way to define the general, that is multivariate normal distribution (including univariate as a special case) is via its characteristic function

We say that a $p$-dimensional random vector $\mathbf{X}$ has **normal distribution**, if and only if its characteristic function is

$$\varphi(\mathbf{t}) = e^{i\mathbf{t}^\top\boldsymbol{\mu} - \frac{1}{2}\mathbf{t}^\top\boldsymbol{\Sigma}\mathbf{t}}$$

for some $p$-dimensional vector $\boldsymbol{\mu}$ and some nonnegative definite $p \times p$ symmetric matrix $\boldsymbol{\Sigma}$; in such a case, we write that $\mathbf{X}$ has the $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ distribution

This definition and notation is justified by the fact that the characteristic function unambiguously determines the probability distribution, and $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ uniquely determine the characteristic function given here

The definition, as given here, allows for singular (degenerate) alternatives; the version precluding those would require that $\boldsymbol{\Sigma}$ is *positive definite*

## Normal distribution: univariate case

In the univariate case, normal distribution is uniquely determined by an arbitrary number $\mu$ and an arbitrary nonnegative number $\sigma^2$. The characteristic function of normal distribution reduces to

$$\varphi(t) = e^{it\mu - \frac{1}{2}t^2\sigma^2}$$

For $\sigma^2 > 0$ it is the characteristic function of the distribution with the density

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

(This can be shown in a slightly involved, but yet accessible way)

The singular (degenerate) cases are those with $\sigma^2 = 0$; in such case there is no density, as $P[X = \mu] = 1$. These cases can be viewed as certain limits when $\sigma^2 \to 0$ for fixed $\mu$

Also, it can be shown via direct calculation with characteristic function that if $X$ has univariate normal distribution with parameters $\sigma^2$ and $\mu$, then $aX + b$ has also univariate normal distribution with respective parameters $(a\sigma)^2$ and $a\mu + b$

## Normal distribution: moments - univariate

The general characteristic function of normal distribution has continuous derivatives of all orders, which implies the existence of all moments. Thus a random vector $\mathbf{X}$ with normal distribution has well-defined and finite $\mathsf{E}(\mathbf{X})$ and $\mathsf{Var}(\mathbf{X})$

In the univariate case, when a random variable $X$ has univariate normal distribution, then the differentiation of its characteristic function at 0 (or direct calculation) yields

$$\mathsf{E}(X) = \mu$$

$$\mathsf{Var}(X) = \sigma^2$$

This includes also singular situations and is consistent with the distribution of the transformed variable $aX + b$, as

$$\mathsf{E}(aX + b) = a\mu + b \qquad \text{and} \qquad \mathsf{Var}(aX + b) = a^2\sigma^2$$

## Normal distribution: general multivariate

Once certain properties of the univariate normal distribution (in particular, the correspondence between the characteristic function and the density, and also the closeness under affine transformations) are established, it is possible to bypass (to an extent) characteristic functions and give an equivalent characterization (and thus an equivalent definition) of the (multivariate) normal distribution via the distributions of linear combinations.

A random vector $\mathbf{X}$ has (multivariate) normal distribution, if and only if any linear combination, $\boldsymbol{\ell}^{\top}\mathbf{X}$, of the components of $\mathbf{X}$ has univariate normal distribution

This property immediately implies the transformation property for general normal distribution: if a random vector $\mathbf{X}$ has normal distribution, then also a vector $\mathbf{AX} + \mathbf{b}$ has normal distribution, for arbitrary $\mathbf{A}$ and $\mathbf{b}$. Thus, if a vector has normal distribution, then its every subvector (created by taking certain components, not necessarily adjacent ones) has normal distribution; and in particular, every component has normal distribution

## Normal distribution: moments - general

The transformation property then also implies the existence of the first and second moments. Let the vector $\boldsymbol{\ell}_i$ has $i$-th component equal to 1 and all others to 0

If a random vector $\mathbf{X}$ has normal distribution, then all its components $X_i = \boldsymbol{\ell}_i^\top \mathbf{X}$ have univariate normal distributions, with means $\mu_i$; thus, we can introduce $\boldsymbol{\mu}$ to denote the mean of the general normal distribution

$$\mathsf{E}(\mathbf{X}) = (\mu_1, \mu_2, \ldots, \mu_p)^\top = \boldsymbol{\mu}$$

The existence of all componentwise variances yields, via the Cauchy-Schwarz inequality, the existence of $\mathsf{Cov}(\boldsymbol{\ell}_i^\top \mathbf{X}, \boldsymbol{\ell}_j^\top \mathbf{X})$ for all $i$ and $j$ - these covariances form the corresponding elements of the variance-covariance matrix $\mathsf{Var}(\mathbf{X})$; the latter is to be denoted by $\boldsymbol{\Sigma}$

Given all this, one can conclude that if $\mathbf{X}$ has a general normal distribution with mean $\boldsymbol{\mu}$ and variance-covariance matrix $\boldsymbol{\Sigma}$, then $\boldsymbol{\ell}^\top \mathbf{X}$, for any $\boldsymbol{\ell}$, has univariate normal distribution with mean $\boldsymbol{\ell}^\top \boldsymbol{\mu}$ and variance $\boldsymbol{\ell}^\top \boldsymbol{\Sigma} \boldsymbol{\ell}$

## Normal distribution: proof of the characterization

Once again: after defining univariate normal distribution and establishing some of its properties, we can define the general normal distribution by the property that any linear combination, $\boldsymbol{\ell}^\mathsf{T}\mathbf{X}$, of the components of $\mathbf{X}$ has univariate normal distribution.

The fact that the distribution of $\mathbf{X}$ is completely characterized by the collection of one-dimensional distributions of $\boldsymbol{\ell}^\mathsf{T}\mathbf{X}$ is the Cramér-Wold theorem. Its proof is also via characteristic functions:

$$\text{if } \mathsf{Y} = \ell^\mathsf{T}\mathbf{X} \quad \text{then} \qquad \varphi_\mathsf{Y}(s) = \mathsf{E}(\mathrm{e}^{\mathrm{i}s\mathsf{Y}}) = \mathsf{E}(\mathrm{e}^{\mathrm{i}s\ell^\mathsf{T}\mathbf{x}})$$

and for $s = 1$, $\varphi_y(1) = \mathsf{E}(\mathrm{e}^{\mathrm{i}\ell^\mathsf{T}\mathbf{Y}})$, which, regarded as a function of $\ell$, is a characteristic function of the random vector $\mathbf{X}$.

## Normal distribution: characterization proof continued

Having defined general normal distribution via the linear combinations, we can derive several properties of it, as shown above. At some point, however, we need the characteristic function of the general normal distribution: we derive it from that of the univariate one. In that way, we prove the "only if" part of the characterization of the normal distribution given above; the "if" part follows by the manipulations with general characteristic function

Knowing that $Y = \ell^\top \mathbf{X}$ has normal distribution with mean $\boldsymbol{\ell}^\top \boldsymbol{\mu}$ and variance $\boldsymbol{\ell}^\top \boldsymbol{\Sigma} \boldsymbol{\ell}$, we obtain the characteristic function of $Y$ as

$$\varphi_Y(t) = e^{it\boldsymbol{\ell}^\top \boldsymbol{\mu} - \frac{1}{2}t^2 \boldsymbol{\ell}^\top \boldsymbol{\Sigma} \boldsymbol{\ell}}$$

This yields

$$\varphi_{\mathbf{X}}(\boldsymbol{\ell}) = \mathsf{E}(e^{i\boldsymbol{\ell}^\top \mathbf{X}}) = \mathsf{E}(e^{iY}) = \varphi_Y(1) = e^{i\boldsymbol{\ell}^\top \boldsymbol{\mu} - \frac{1}{2}\boldsymbol{\ell}^\top \boldsymbol{\Sigma} \boldsymbol{\ell}}$$

- which is identical to the characteristic function introduced at the beginning

## Normal distribution and independence

If $\mathbf{X}_1$ is $\mathrm{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$, $\mathbf{X}_2$ is $\mathrm{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$, and they are independent

then $\mathbf{X} = \begin{pmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \end{pmatrix}$ is $\mathrm{N}\left(\begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{pmatrix} \begin{pmatrix} \boldsymbol{\Sigma}_1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_2 \end{pmatrix}\right)$

Conversely, if $\mathbf{X}$ has normal distribution, and the structure of the variance-covariance matrix is as above - which mean that $\mathbf{X}_1$ and $\mathbf{X}_2$ are uncorrelated - then not only $\mathbf{X}_1$ and $\mathbf{X}_2$ both have normal distribution, but they are also *independent*

Both of these assertions can be now verified by manipulations with characteristic functions - and obviously, they are valid not only for two, but for any finite collection of random vectors $\mathbf{X}$

Yes, and there are some finessess here:

Suppose that $\mathbf{X}_1$ has normal distribution, and $\mathbf{X}_2$ has normal distribution. Does the vector $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2)^{\mathsf{T}}$ have normal distribution (we may say then that $\mathbf{X}_1$ and $\mathbf{X}_2$ are *jointly* normal, to emphasize that)? Well, if $\mathbf{X}_1$ and $\mathbf{X}_2$ are independent, then the property above says that yes, but otherwise it may be one way or another...

## Normal distribution has a density...

The independence property immediately yields that if $\mathbf{U}$ is a random vector consisting of $p$ independent components $U_i$, each with a standard normal distribution $N(0,1)$, then $\mathbf{U}$ has the $N(\mathbf{0},\mathbf{I})$ normal distribution. Its density is the product of $p$ standard normal densities

$$g(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^p}} e^{-\frac{1}{2}\mathbf{x}^\top \mathbf{x}}$$

In the general case, variance-covariance matrix $\boldsymbol{\Sigma}$ is a symmetric matrix: thus there is a matrix $\boldsymbol{\Sigma}^{1/2}$ such that $\boldsymbol{\Sigma}^{1/2}\boldsymbol{\Sigma}^{1/2} = \boldsymbol{\Sigma}$

In the *nonsingular*, non-degenerate case, when the variance-covariance matrix $\boldsymbol{\Sigma}$ is positive definite and thus invertible, then $\boldsymbol{\Sigma}^{1/2}$ is invertible too, and the density transformation theorem says that the distribution of $\mathbf{X} = \boldsymbol{\Sigma}^{1/2}\mathbf{U} + \boldsymbol{\mu}$ - which the examination of moments shows to be $N(\boldsymbol{\mu},\boldsymbol{\Sigma})$ - has the density

$$f(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^p \det(\boldsymbol{\Sigma})}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}$$

## ... if its variance-covariance matrix is nonsingular

So, in the nonsingular, non-degenerate case, when the variance-covariance matrix $\mathbf{\Sigma}$ is invertible and thus positive definite, normal distribution $\mathrm{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$ has a density

If $\mathbf{\Sigma}$ is singular, then $\mathbf{X}$ does not have a density with respect to the Lebesgue measure on $\mathbb{R}^p$, but it may have one when restricted to some affine subspace of $\mathbb{R}^p$, with respect to some appropriately-dimensional Lebesgue measure on that subspace

## Summary

Let us summarize the most important properties now:

- Normal is characterized by first and second moments

  by $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$

- Affine transformation of normal is normal

  and then it comes again to first and second moments:

  if $\mathbf{X}$ is $\mathrm{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, then $\mathbf{AX} + \mathbf{b}$ is $\mathrm{N}(\mathbf{A}\boldsymbol{\mu} + \mathbf{b}, \mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^\top)$

and in particular, all subvectors are normal

- If subvectors of normal are uncorrelated, they are independent

  ("uncorrelated" means zeros in the appropriate part of $\boldsymbol{\Sigma}$)

- Normal has a density $\quad f(\mathbf{x}) = \dfrac{1}{\sqrt{(2\pi)^p \det(\boldsymbol{\Sigma})}} \mathrm{e}^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}$

  but only if $\boldsymbol{\Sigma}$ is nonsingular, positive definite

And here again beware: random variable $X$ has normal distribution, random variable $Y$ has normal distribution, they are uncorrelated: are they independent?

## From Univariate to Multivariate Normal Sampling: One-Sample Hotelling's $T^2$

## Normal sampling: recall univariate setting first

We have $n$ numbers, $y_1, y_2, \ldots, y_n$ - and *we believe that they can be modeled as the outcomes* of $n$ random variables $Y_1, Y_2, \ldots, Y_n$ - which are iid with the distribution $N(\mu, \sigma^2)$

(iid = "independent identically distributed")

In such case, we have (for all $i$) $\mu = \mathsf{E}(X_i)$

and $\sigma^2 = \mathsf{Var}(Y_i) = \mathsf{E}[((Y_i) - \mathsf{E}(Y_i))^2]$

Statistics then suggests to estimate

$$\mu \quad \text{by} \quad \bar{y} = \frac{1}{n}\sum_i y_i$$

$$\text{and} \quad \sigma^2 \quad \text{by} \quad s^2 = \frac{1}{n-1}\sum_i (y_i - \bar{y})^2$$

The subsequent mathematics then yields $\mathsf{Var}(\bar{Y}) = \dfrac{1}{n}\sigma^2$

which statistics suggests to be estimated by $\dfrac{1}{n}s^2$

## Multivariate normal sampling: assumptions

So, we have $\mathbf{Y}$ arising by iid sampling - recall:

The data matrix $\mathbf{Y}$, formed by $y_{ij}$

has $n$ lines, $i = 1, \ldots, n$

of $p$ observations, $j = 1, \ldots, p$

which we model via $n$ (column) random vectors $\mathbf{y}_1, \mathbf{y}_2, \ldots \mathbf{y}_n$

such that outcomes of $\mathbf{y}_1^T, \mathbf{y}_2^T, \ldots \mathbf{y}_n^T$ give rows of $\mathbf{Y}$

these $\mathbf{y}_i$'s have each $p$ components, and are independent

that is, independent between themselves

(but not necessarily within, between their components)

and each has the distribution of some generic random vector $\mathbf{y}$

*And this distribution is now* $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

## Multivariate sampling: estimation

If $\mathbf{y} = (Y_1, Y_2, \ldots, Y_P)^\top$ is any of these random vectors, then

$$\boldsymbol{\mu} = \mathrm{E}(\mathbf{y}) = (\mathrm{E}(Y_1), \ldots, \mathrm{E}(Y_p))^\top \quad \text{and} \quad \boldsymbol{\Sigma} = \mathrm{Var}(\mathbf{X})$$

We estimate $\boldsymbol{\mu}$ by the vector of columnwise means

$$\bar{\mathbf{y}} = \left( \frac{1}{n} \sum_i y_{i1}, \ldots, \frac{1}{n} \sum_i y_{ip} \right)^\top$$

and $\boldsymbol{\Sigma}$ by the sample variance-covariance matrix $\mathbf{S}$, with elements

$$S_{k\ell} = \frac{1}{n-1} \sum_i (y_{ik} - \bar{y}_k)(y_{i\ell} - \bar{y}_\ell)$$

Note: in what follows, we will abuse notation, using $\bar{\mathbf{y}}$ for $\dfrac{1}{n}\mathbf{1}^\top\mathbf{Y}$, the (column) vector of columnwise means (a non-random vector) and also for the average $\mathbf{y}_i = \dfrac{1}{n} \sum_i \mathbf{y}_i$ of random vectors (a random vector)

## Multivariate normal sampling: distribution

The sum of independent random vectors with normal distribution is normal; in particular, if $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n$ are iid with $\mathrm{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

$$\text{then } \bar{\mathbf{y}} = \frac{1}{n}\sum_i \mathbf{y}_i \quad \text{is} \quad \mathrm{N}\left(\boldsymbol{\mu}, \frac{1}{n}\boldsymbol{\Sigma}\right)$$

which implies that $\boldsymbol{\Sigma}^{-1/2}(\bar{\mathbf{y}} - \boldsymbol{\mu})\sqrt{n}$ is $\mathrm{N}(\mathbf{0}, \mathbf{I})$

- if $\boldsymbol{\Sigma}$ is nonsingular, of course; then we have a matrix $\boldsymbol{\Sigma}^{-1/2}$ inverse to $\boldsymbol{\Sigma}^{1/2}$ and such that $\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\Sigma}^{-1/2} = \boldsymbol{\Sigma}^{-1}$

If $\boldsymbol{\Sigma}$ is nonsingular, then $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n$ have a density which is a product of individual densities; the maximum likelihood estimates based on their outcomes $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n$ are

$$\widehat{\boldsymbol{\mu}} = \bar{\mathbf{y}} = \frac{1}{n}\sum_i \mathbf{y}_i$$

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{n}\sum_i (\mathbf{y}_i - \bar{\mathbf{y}})(\mathbf{y}_i - \bar{\mathbf{y}})^\top = \frac{n-1}{n}\mathbf{S}$$

where $\mathbf{S}$ is the usual sample variance-covariance matrix

## Chi-square: review, essentially univariate

*From now on, $\mathbf{\Sigma}$ is assumed positive definite - nonsingular, if not stated otherwise*

**Chi-square distribution**

If $\mathbf{x}$ is $N(\mathbf{0}, \mathbf{I})$, then $\mathbf{x}^\top\mathbf{x}$ is $\chi^2_p$ with $p = \dim \mathbf{x}$

Consequently, if $\mathbf{x}_1, \ldots, \mathbf{x}_n$ are iid $N(\boldsymbol{\mu}, \mathbf{\Sigma})$

then $n(\bar{\mathbf{x}} - \boldsymbol{\mu})^\top \mathbf{\Sigma}^{-1}(\bar{\mathbf{x}} - \boldsymbol{\mu})$ is $\chi^2_p$

Asymptotics ($n \to \infty$, $n >> p$)

If $\mathbf{x}_1$, ..., $\mathbf{x}_n$ are iid and $n - p$ is large

then $\sqrt{n}(\bar{\mathbf{x}} - \boldsymbol{\mu})$ is approximately $N(\mathbf{0}, \mathbf{\Sigma})$

and thus $n(\bar{\mathbf{x}} - \boldsymbol{\mu})^\top \mathbf{\Sigma}^{-1}(\bar{\mathbf{x}} - \boldsymbol{\mu})$ is approximately $\chi^2_p$

and thus also $n(\bar{\mathbf{x}} - \boldsymbol{\mu})^\top \mathcal{S}^{-1}(\bar{\mathbf{x}} - \boldsymbol{\mu})$ is approximately $\chi^2_p$

(we may replace $\mathbf{\Sigma}$ by $\mathbf{S}$ if $n - p$ is large)

## Univariate: t and F

**t distribution** “Student’s t”

If $U$ and $V$ are independent, $U$ is $N(0,1)$, $V$ is $\chi^2_k$, then

$$\frac{U}{\sqrt{\frac{V}{k}}} \text{ is } t_k$$

**F distribution** “Fisher’s(-Snedecor’s) F”

If $U$ and $V$ are independent, $U$ is $\chi^2_\ell$, $V$ is $\chi^2_k$, then

$$\frac{\frac{U}{\ell}}{\frac{V}{k}} \quad \text{is} \quad F_{\ell,k}$$

Note: if $Z$ is $t_k$, then $Z^2$ is $F_{1,k}$

## And now: Wishart distribution, univariate prelude

If $X_1, X_2, \ldots, X_n$ are iid $N(\mu, \sigma^2)$

then $\bar{X}$ and $s^2$ are independent

and $\quad (n-1)\dfrac{s^2}{\sigma^2} \quad$ is $\quad \chi^2_{n-1}$

We may think of it in a way that $(n-1)s^2$ is $\sigma^2\chi^2_{n-1}$

where $U$ being $\sigma^2\chi^2_{n-1}$ means $\dfrac{U}{\sigma^2}$ is $\chi^2_{n-1}$

and the distribution of $(n-1)s^2$ is

the distribution of $\displaystyle\sum_{i=1}^{n-1} Z_i^2$, when $Z_i$ are iid $N(0, \sigma^2)$

This is a special case of the Wishart distribution for $\boldsymbol{\Sigma} = (\sigma^2)$, with $n-1$ degrees of freedom

In the multivariate situation, $\mathbf{U}$ and $\boldsymbol{\Sigma}$ are matrices - so it is not that straightforward to divide one by another; $\mathbf{U}\boldsymbol{\Sigma}^{-1}$ may work sometimes, but sometimes not

## Wishart distribution: definition (full force)

Suppose that random vectors $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_m$ are iid with $N(\mathbf{0}, \boldsymbol{\Sigma})$

The distribution of $\sum_{i=1}^{m} \mathbf{z}_i \mathbf{z}_i^\top$ is called

**Wishart distribution** with parameters

$m$ (degrees of freedom, positive integer), and

$\boldsymbol{\Sigma}$ (any nonnegative definite symmetric matrix)

Notation: $\mathcal{W}_m(\boldsymbol{\Sigma})$ (it is often assumed that $\boldsymbol{\Sigma}$ is *positive* definite)

Also note: if $\mathbf{Y}$ is an $m \times p$ matrix whose *rows* are $\mathbf{z}_i^\top$

then $\sum_{i=1}^{m} \mathbf{z}_i \mathbf{z}_i^\top = \mathbf{Y}^\top \mathbf{Y}$

If $\mathbf{L}$ is a diagonal matrix with diagonal elements $\ell_i$

then $\mathbf{Y}^\top \mathbf{L} \mathbf{Y} = \sum_{i=1}^{m} \ell_i \mathbf{z}_i \mathbf{z}_i^\top$

## Wishart distribution: first properties

If $\mathbf{B}$ has $\mathcal{W}_m(\mathbf{\Sigma})$ distribution

and $\mathbf{A}$ is a fixed (non-random), general $q \times p$ matrix,

then $\mathbf{A}^\top\mathbf{B}\mathbf{A}$ has $\mathcal{W}_m(\mathbf{A}^\top\mathbf{\Sigma}\mathbf{A})$ distribution

If $\mathbf{B}_1$ has $\mathcal{W}_{m_1}(\mathbf{\Sigma})$ and $\mathbf{B}_2$ has $\mathcal{W}_{m_2}(\mathbf{\Sigma})$ distribution,

and $\mathbf{B}_1$ and $\mathbf{B}_2$ are independent,

then $\mathbf{B}_1 + \mathbf{B}_2$ has $\mathcal{W}_{m_1+m_2}(\mathbf{\Sigma})$ distribution

If $\mathbf{B}$ has $\mathcal{W}_m(\mathbf{\Sigma})$ distribution

and $\mathbf{a}$ is an arbitrary fixed (non-random) vector, $\mathbf{a}^\top\mathbf{\Sigma}\mathbf{a} \neq 0$,

then $\dfrac{\mathbf{a}^\top\mathbf{B}\mathbf{a}}{\mathbf{a}^\top\mathbf{\Sigma}\mathbf{a}}$ has $\chi^2_m$ distribution

(which is then independent of $\mathbf{a}$)

(This is how the division of $\mathbf{U}$ by $\mathbf{\Sigma}$ may eventually fly)

## Wishart distribution: the important property

Suppose that $\mathbf{Y}$ is an $n \times p$ matrix whose rows are $\mathbf{y}_i^\top$,
where $\mathbf{y}_i$ are iid random vectors with distribution $N(\mathbf{0}, \boldsymbol{\Sigma})$;
if $\mathbf{P}$ is a symmetric *idempotent* matrix,
then $\mathbf{Y}^\top\mathbf{P}\mathbf{Y}$ has $W_m(\boldsymbol{\Sigma})$ distribution,
where $m$ is equal to the trace (rank) of $\mathbf{P}$

As a consequence, if $\mathbf{S}_\mathbf{Y}$ is the sample variance-covariance matrix
calculated out of $\mathbf{Y}_i$,
then the distribution of $(n-1)\mathbf{S}_\mathbf{Y}$ is $W_{n-1}(\boldsymbol{\Sigma})$

## Wishart distribution: one more property

For $m > p$ and arbitrary $p \times p$ symmetric, nonnegative definite $\mathbf{\Sigma}$,

$\mathcal{W}_m(\mathbf{\Sigma})$ is (absolutely) continuous (and thus has a density)

If $\mathbf{B}$ has $\mathcal{W}_m(\mathbf{\Sigma})$ distribution

and $\mathbf{a}$ is an arbitrary fixed (non-random) vector,

then $\dfrac{\mathbf{a}^\top \mathbf{\Sigma}^{-1} \mathbf{a}}{\mathbf{a}^\top \mathbf{B}^{-1} \mathbf{a}}$ has $\chi^2_{m-p+1}$ distribution

(which is then independent of $\mathbf{a}$)

(This is how the division of $\mathbf{U}$ by $\mathbf{\Sigma}$ is done in sampling...)

## An important corollary

If $\mathbf{d}$ and $\mathbf{B}$ are independent, $\mathbf{d}$ is $N(\mathbf{0}, \mathbf{\Sigma})$, $\mathbf{B}$ is $W_m(\mathbf{\Sigma})$, then

$$\frac{m-p+1}{p}\,\mathbf{d}^\top\mathbf{B}^{-1}\mathbf{d} = \frac{\dfrac{\mathbf{d}^\top\mathbf{\Sigma}^{-1}\mathbf{d}}{p}}{\dfrac{\dfrac{\mathbf{d}^\top\mathbf{\Sigma}^{-1}\mathbf{d}}{\mathbf{d}^\top\mathbf{B}^{-1}\mathbf{d}}}{m-p+1}} \qquad \text{which is} \qquad \frac{\dfrac{\chi^2_p}{p}}{\dfrac{\chi^2_{m-p+1}}{m-p+1}}$$

- well, the latter is not a rigorous math formula, only some shorthand, memory aid, for expressing that if $U$ has $\chi^2_p$, $V$ has $\chi^2_{m-p+1}$ and $U$ and $V$ are independent (do not forget the latter!), and that the expression above is equal to

$$\frac{\dfrac{U}{p}}{\dfrac{V}{m-p+1}} \qquad \text{which has the same distribution as} \qquad \frac{m-p+1}{p}\,\mathbf{d}^\top\mathbf{B}^{-1}\mathbf{d}$$

- and thus the latter has distribution $F_{p,m-p+1}$

## Univariate sampling once again...

We have $n$ numbers, $y_1, y_2, \ldots, y_n$

they can be modeled as the outcomes of $n$ random variables f $Y_1, Y_2, \ldots, Y_n$ are iid with the distribution $N(\mu, \sigma^2)$

In such case, we have (for all $i$) $\mu = E(Y_i)$

and $\sigma^2 = \mathrm{Var}(Y_i) = E[((Y_i) - E(Y_i))^2]$

Statistics then suggests to estimate

$$\mu \quad \text{by} \quad \bar{y} = \frac{1}{n}\sum_i y_i$$

$$\text{and} \quad \sigma^2 \quad \text{by} \quad s^2 = \frac{1}{n-1}\sum_i (y_i - \bar{y})^2$$

The subsequent mathematics then yields $\mathrm{Var}(\bar{Y}) = \dfrac{1}{n}\sigma^2$

which statistics suggests to be estimated by $\dfrac{1}{n}s^2$

and also the standard deviation of $\bar{Y}$, $\dfrac{1}{\sqrt{n}}\sigma$, by $\dfrac{1}{\sqrt{n}}s$

## … concluded by the t-ratio (univariate)

t-ratio: $$\frac{\bar{Y}-\mu}{\frac{s}{\sqrt{n}}} = \sqrt{n}\,\frac{\bar{Y}-\mu}{s}$$ (“Student’s t-ratio”)

has distribution $t_{n-1}$, that is, $\dfrac{N(0,1)}{\sqrt{\frac{\chi^2_{n-1}}{n-1}}}$ (symbolically)

which may be approximated by $N(0,1)$ for large $n$

*And now*: t-ratio squared: $$n\,\frac{(\bar{Y}-\mu)^2}{s^2}$$

has distribution $F_{1,n-1}$, that is $\dfrac{\frac{\chi^2_1}{1}}{\frac{\chi^2_{n-1}}{n-1}}$ (again symbolically)

and may be for large $n$ approximated by $\chi^2_1$

symbolically: $N(0,1)^2$

## Recall multivariate normal sampling now

Now, we have an array of numbers $y_{ij}$, forming the data matrix $\mathbf{Y}$

- $n$ vectors, $i = 1, \ldots, n$, of $p$ observations, $j = 1, \ldots, p$

We model them via $n$ random vectors $\mathbf{y}_1, \mathbf{y}_2, \ldots \mathbf{y}_n$

those that give rows of $\mathbf{Y}$ as their outcomes

each with $p$ components, and independent

that is, between themselves

(but not necessarily within, between their components)

with the same distribution $\mathrm{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

If $\mathbf{y}$ is any of these random vectors,

then we have $\boldsymbol{\mu} = \mathsf{E}(\mathbf{y})$ and $\boldsymbol{\Sigma} = \mathsf{Var}(\mathbf{y})$

## And then

We estimate $\boldsymbol{\mu}$ by the vector of the columnwise means

as we know that $\bar{\mathbf{y}} = \frac{1}{n}\sum_{i-1}^{n} \mathbf{y}_i$ has distribution $N\left(\boldsymbol{\mu}, \frac{1}{n}\boldsymbol{\Sigma}\right)$

Then we estimate $\boldsymbol{\Sigma}$ by the sample variance-covariance matrix $\mathbf{S}$,

as we know that $(n-1)\mathbf{S}$ has distribution $\mathcal{W}_{n-1}(\boldsymbol{\Sigma})$

And after we observe that $(n-1)\mathbf{S}$ and $\bar{\mathbf{y}}$ are independent

We form not the t-ratio, but its square

## t-ratio (squared): multivariate (Hotelling's $\mathsf{T}^2$)

$\mathsf{T}^2 = n(\bar{\mathbf{y}} - \boldsymbol{\mu})^\top \mathbf{S}^{-1}(\bar{\mathbf{y}} - \boldsymbol{\mu})$ which has

(under all the assumptions above)

the distribution $\dfrac{(n-1)p}{n-p}\mathsf{F}_{p,n-1-p+1} = \dfrac{(n-1)p}{n-p}\mathsf{F}_{p,n-p}$,

which actually means that

$$\frac{n-p}{(n-1)p}\mathsf{T}^2 = \frac{n-p}{(n-1)p}n(\bar{\mathbf{y}} - \boldsymbol{\mu})^\top \mathbf{S}^{-1}(\bar{\mathbf{y}} - \boldsymbol{\mu})$$

has distribution $\mathsf{F}_{p,n-p}$

(which for large $n$ may be approximated by $\chi^2_p$)

Note: for $p = 1$ the first fraction is 1 and the distribution is $\mathsf{F}_{1,n-1}$ - same as for the univariate squared t-ratio.

Once again: we keep abusing the notation, using sometimes $\bar{\mathbf{y}}$ for the (column) vector of columnwise means, $\dfrac{1}{n}\mathbf{1}^\top \mathbf{Y}$ (a non-random vector), and sometimes for the average of random vectors $\mathbf{y}_i$ (a random vector) - depending on a context

## First applications

First applications are sort of analogs of univariate ones: we can

**test a hypothesis**

$$H_0\colon \boldsymbol{\mu} = \boldsymbol{\mu}_0$$

using $\quad T^2 = n(\bar{\mathbf{y}} - \boldsymbol{\mu}_\mathbf{0})^\top \mathbf{S}^{-1}(\bar{\mathbf{y}} - \boldsymbol{\mu}_\mathbf{0}) \quad$ (substitute $\boldsymbol{\mu} = \boldsymbol{\mu}_0$)

rejecting $H_0$ if $T^2 > \dfrac{(n-1)p}{n-p} F_{p,n-p}(\alpha)$

In the similar vein, we can form

**confidence sets**

$\boldsymbol{\mu}$ in $T^2$ becomes a variable

yielding a confidence set which is an ellipsoid such that

$$P\Big[(\bar{\mathbf{y}} - \boldsymbol{\mu})^\top \mathbf{S}^{-1}(\bar{\mathbf{y}} - \boldsymbol{\mu}) \leqslant \frac{(n-1)p}{n(n-p)} F_{p,n-p}(\alpha)\Big] = 1 - \alpha.$$

## Paired comparisons

This is still an application in the univariate style: a paired t-test is generalized to the situation when there are not pairs of one, but several variables (the pairing is one-to-one)

The observations come as are $n$ pairs of vectors $\mathbf{y}_{1i}$ and $\mathbf{y}_{2i}$

similarly to the univariate, classical paired t-test

$\mathbf{y}_{1i}$ and $\mathbf{y}_{2i}$ are *jointly normal*

not necessary independent within or between themselves

but *pairs of vectors* are independent for different $i = 1, 2, \ldots, n$

and have the same distribution (independent of $i = 1, 2, \ldots, n$)

In such case, the (vector) differences $\mathbf{z}_i = \mathbf{y}_{1i} - \mathbf{y}_{2i}$ can be modeled

by iid random vectors with normal distribution $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

The hypothesis $H_0 : \boldsymbol{\mu} = \mathbf{0}$ is then rejected if

$$T^2 = n\bar{\mathbf{z}}^\mathsf{T}\mathbf{S}^{-1}\bar{\mathbf{z}} > \frac{(n-1)p}{n-p} F_{p,n-p}(\alpha)$$

## And... a Technology of Its Own: Repeated Measures

## Truly multivariate application: repeated measures

In a sense, it is a generalization of paired t-test again, but in different direction: now we do not compare two paired groups of vectors, but do it in a more involved way

We do it with a data matrix $\mathbf{Y}$ using (the matrix of) **contrasts**, believing, as before, that the rows of $\mathbf{Y}$ are realizations of iid random vectors $\mathbf{y}_i$

A **contrast matrix** is defined to be any $k \times p$ matrix $\mathbf{C}$ that has sums of rows equal to zero; the comparisons we are after are expressed through $\mathbf{C}\mathbf{y}_i$

Note: such a contrast matrix may contain maximally $p-1$ linearly independent lines (why $p-1$ and not $p$? because of the summation to 0 requirement)

Let us have a look at some examples with $p = 4$

## Linear models in R recalled: default

```
> sk=expand.grid(c("a","b","b","a"),c("a","b","a","b"))
> sk[,3] = rnorm(16)
> summary(lm(V3~Var1*Var2,data=sk))
...
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.4713     0.5238  -0.900    0.386
Var1b         1.0322     0.7408   1.393    0.189
Var2b         0.5817     0.7408   0.785    0.448
Var1b:Var2b  -1.0535     1.0476  -1.006    0.334
...
> dummy.coef(lm(V3~Var1*Var2,data=sk))
Full coefficients are

(Intercept):    -0.4713099
Var1:                    a         b
                  0.000000  1.032178
Var2:                    a         b
                 0.0000000 0.5816988
Var1:Var2:             a:a       b:a       a:b       b:b
                  0.000000  0.000000  0.000000 -1.053515
```

## Linear models in R recalled: “theoretical default”

```
> options(contrasts=c("contr.sum","contr.poly"))
> summary(lm(V3~Var1*Var2,data=sk))
...
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.07225    0.26191   0.276    0.787
Var11       -0.25271    0.26191  -0.965    0.354
Var21       -0.02747    0.26191  -0.105    0.918
Var11:Var21 -0.26338    0.26191  -1.006    0.334
...
> dummy.coef(lm(V3~Var1*Var2,data=sk))
Full coefficients are

(Intercept):     0.07224971
Var1:                     a          b
                 -0.2527102  0.2527102
Var2:                     a          b
                -0.02747062 0.02747062
Var1:Var2:              a:a        b:a        a:b        b:b
                 -0.2633788  0.2633788  0.2633788 -0.2633788
```

## Examples of contrasts that carry special meaning

Compare all to a control:

```
> t(contr.sum(4))
     1 2 3  4
[1,] 1 0 0 -1
[2,] 0 1 0 -1
[3,] 0 0 1 -1
```

Compare first to second, second to third, third to fourth,...:

```
> t(array(dim=c(4,3),c(1,-1,0,0,0,1,-1,0,0,0,1,-1)))
     [,1] [,2] [,3] [,4]
[1,]    1   -1    0    0
[2,]    0    1   -1    0
[3,]    0    0    1   -1
```

Of course, we do not have to have $k = 3$ lines in each contrast; we can consider contrasts also with $k = 1$ or $k = 2$ - but we know $k = p - 1$ is the maximal possible number of linearly independent lines in a contrast; in cases like above, the dependent lines could be identified by simple logic

# More elaborate contrasts

Let a and A be two levels of the first factor, b and B two levels of the seond factor, and the observations at every item correspond to ab, aB, Ab, AB. Then the first contrast evaluates the effect of the first factor, the second of the second factor, and the third one the interaction:

```
> t(array(dim=c(4,3),c(1,1,-1,-1,1,-1,1,-1,1,-1,-1,1)))
     [,1] [,2] [,3] [,4]
[1,]    1    1   -1   -1
[2,]    1   -1    1   -1
[3,]    1   -1   -1    1
```

Such an elaborate contrast is possible for $p = 4$ (or similar)

And again, the lines are independent

Incidentally, do we need independent lines in a contrast matrix?

Well, we will see that it is good to have it like that

## Distribution of repeated measures

Once again, data matrix $\mathbf{Y}$ is assumed to arise from random normal sampling: its lines are modeled as outcomes of (column) iid random vectors $\mathbf{y}_i$ with normal distribution $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

then $\mathbf{z}_i = \mathbf{C}\mathbf{y}_i$ can be modeled as iid random vectors

with normal distribution $N(\mathbf{C}\boldsymbol{\mu}, \mathbf{C}\boldsymbol{\Sigma}\mathbf{C}^\top)$

and we can use again the distributional mathematics (abusing once again the notation and using $\bar{\mathbf{y}}$ also for columnwise means of $\mathbf{Y}$ made into a column vector) for

$$T^2 = n\bar{\mathbf{z}}^\top \mathbf{S}^{-1}_{\mathbf{Y}\mathbf{C}^\top}\bar{\mathbf{z}} = n(\mathbf{C}\bar{\mathbf{y}})^\top(\mathbf{C}\mathbf{S}_\mathbf{Y}\mathbf{C}^\top)^{-1}(\mathbf{C}\bar{\mathbf{y}})$$

concluding that $T^2$ has the distribution $\dfrac{(n-1)k}{n-k}F_{k,n-k}$

Typically, we can test the hypothesis that all contrasts are zero but also other hypothesis, for confidence sets, etc.

And it is all pretty much nothing else but applying the $T^2$: not necessarily to the original variables, but rather to the specific linear combinations of those

So, do we see now why we need $\mathbf{C}$ to have independent lines?

## It is all in the formula for $T^2$

The formula for $T^2$ calls for the invertibility of $\mathbf{CSC}^\top$; as this matrix has dimension $k \times k$, its rank must be $k$ then

Now, the rank of a product of matrices is at most the maximum or ranks of the matrices involved; consequently, the rank of $\mathbf{CSC}^\top$ is not greater than the rank of $\mathbf{S}$ - so if we assume it is $p$, then $k$ cannot be greater than $p$. And the condition that rows sum to zero - in matrix terms expressible as $\mathbf{1}^\top\mathbf{C} = \mathbf{0}$ further drives rank one more down: that is, $k$ cannot be greater than $p-1$, otherwise $\mathbf{CSC}^\top$ cannot have rank $k$

Of course, if $k = p - 1$, it is not guaranteed mathematically that the rank is $k$; but from the statistical point of view, we can always believe that elements of $\mathbf{S}$ are sufficiently "in a general position"

(Try generating a random matrix with components that come from independent realizations of a distribution with a density; the result is then invertible with probability one - and indeed in R, you likely never encounter a noninvertible matrix this way

```
> solve(matrix(rnorm(100),10,10)) # repeat or alter if you wish
...
```

## First, recall again: the data

```
> read.table("electro.d",header=T,row.names=1)
    E1   E2   E3  E4  E5
1  500  400   98 200 250
2  660  600  600  75 310
3  250  370  220 250 220
4   72  140  240  33  54
5  135  300  450 430  70
6   27   84  135 190 180
7  100   50   82  73  78
8  105  180   32  58  32
9   90  180  220  34  64
10 200  290  320 280 135
11  15   45   75  88  80
12 160  200  300 300 220
13 250  400   50  50  92
14 170  310  230  20 150
15  66 1000 1050 280 220
16 107   48   26  45  51
```

We want to compare E1, E2, E3, E4, E5 - somehow in total. Lines are related to a certain subject, hence independence within a line may not be realistic; but independence between lines perhaps yes

## First taking logs (more conforming to normal)

```
> elel=log(read.table("electro.d",header=T,row.names=1))
> elel
         E1       E2       E3       E4       E5
1  6.214608 5.991465 4.584967 5.298317 5.521461
2  6.492240 6.396930 6.396930 4.317488 5.736572
3  5.521461 5.913503 5.393628 5.521461 5.393628
4  4.276666 4.941642 5.480639 3.496508 3.988984
5  4.905275 5.703782 6.109248 6.063785 4.248495
6  3.295837 4.430817 4.905275 5.247024 5.192957
7  4.605170 3.912023 4.406719 4.290459 4.356709
8  4.653960 5.192957 3.465736 4.060443 3.465736
9  4.499810 5.192957 5.393628 3.526361 4.158883
10 5.298317 5.669881 5.768321 5.634790 4.905275
11 2.708050 3.806662 4.317488 4.477337 4.382027
12 5.075174 5.298317 5.703782 5.703782 5.393628
13 5.521461 5.991465 3.912023 3.912023 4.521789
14 5.135798 5.736572 5.438079 2.995732 5.010635
15 4.189655 6.907755 6.956545 5.634790 5.393628
16 4.672829 3.871201 3.258097 3.806662 3.931826
```

## Contrasts and their transposes

Now, contrasts were defined to be $k \times p$ matrices, with each row summing to 0 - that was convenient in the stochastic assumptions, where observations were modeled as column $p \times 1$ vectors.

With an $n \times p$ data matrix **Y**, these observations correspond to its *rows*. Hence we have to multiply **Y** from right right by a $p \times k$ *transposed* contrast - a matrix that has *columns*, not rows, now summing to zero.

In what follows, we show these contrasts as they appear in R - that is, in their transposed form

## Once again some of the examples transposed

Compare all to control

```
> contr.sum(4)
  [,1] [,2] [,3]
1    1    0    0
2    0    1    0
3    0    0    1
4   -1   -1   -1
```

Compare first to second, second to third, third to fourth,...

```
> array(dim=c(4,3),c(1,-1,0,0,0,1,-1,0,0,0,1,-1))
     [,1] [,2] [,3]
[1,]    1    0    0
[2,]   -1    1    0
[3,]    0   -1    1
[4,]    0    0   -1
```

## More elaborate contrasts again

Let a and A be two levels of the first factor, b and B two levels of the seond factor, and the observations at every item correspond to ab, aB, Ab, AB. Then the first contrast (first column) evaluates the effect of the first factor, the second (column) of the second factor, and the third one the interaction (between those):

```
> array(dim=c(4,3),c(1,1,-1,-1,1,-1,1,-1,1,-1,-1,1))
     [,1] [,2] [,3]
[1,]    1    1    1
[2,]    1   -1   -1
[3,]   -1    1   -1
[4,]   -1   -1    1
```

## And now: the analysis of (logarithms of) electrodes

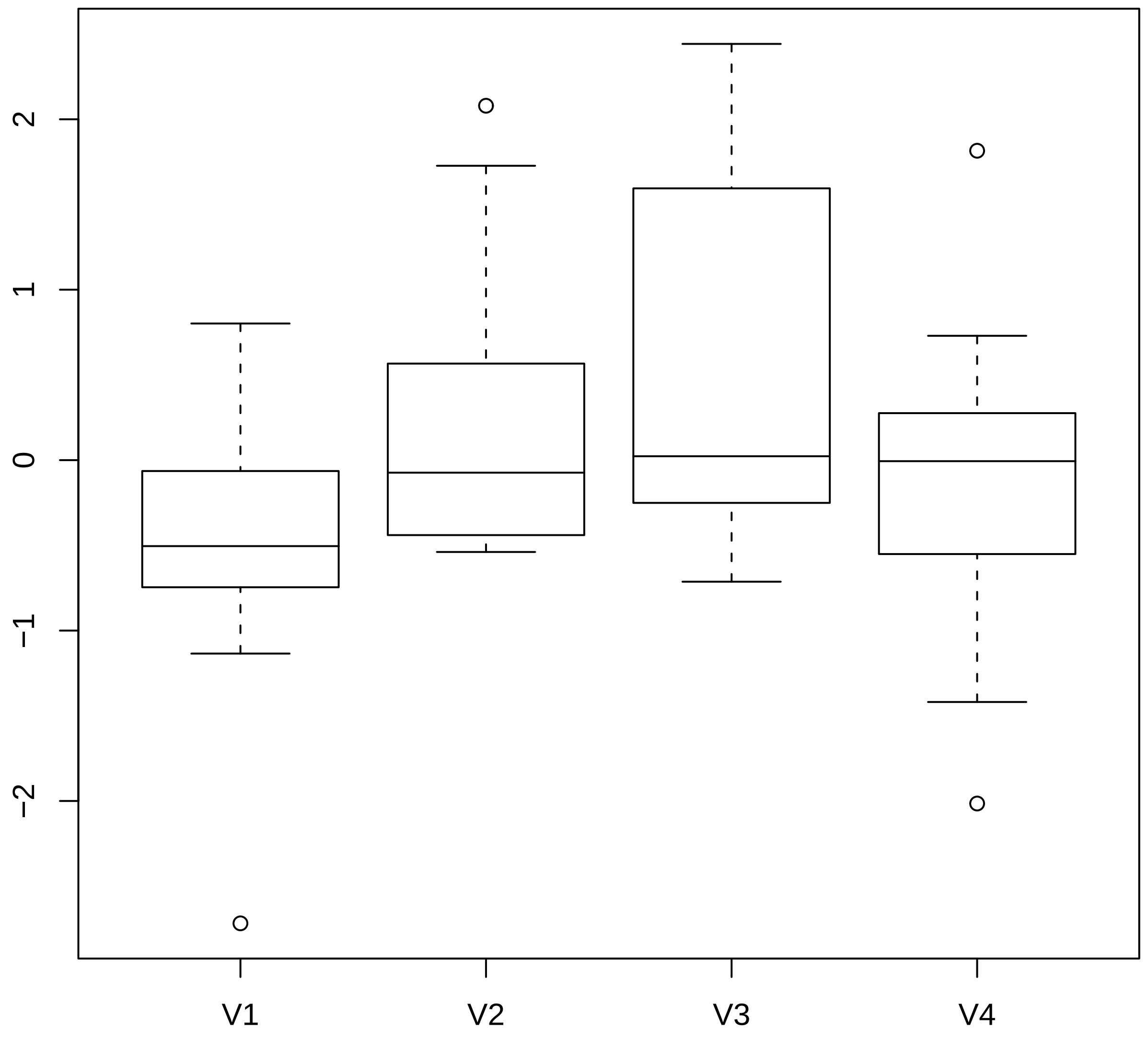
A potentially useful contrast matrix:

```
> elec = array(dim=c(5,4),
+ c(1,-1,0,0,0,0,1,-1,0,0,0,0,1,-1,0,0,0,0,1,-1))
> elec
     [,1] [,2] [,3] [,4]
[1,]    1    0    0    0
[2,]   -1    1    0    0
[3,]    0   -1    1    0
[4,]    0    0   -1    1
[5,]    0    0    0   -1
> elelc=as.matrix(elel) %*% elec
```

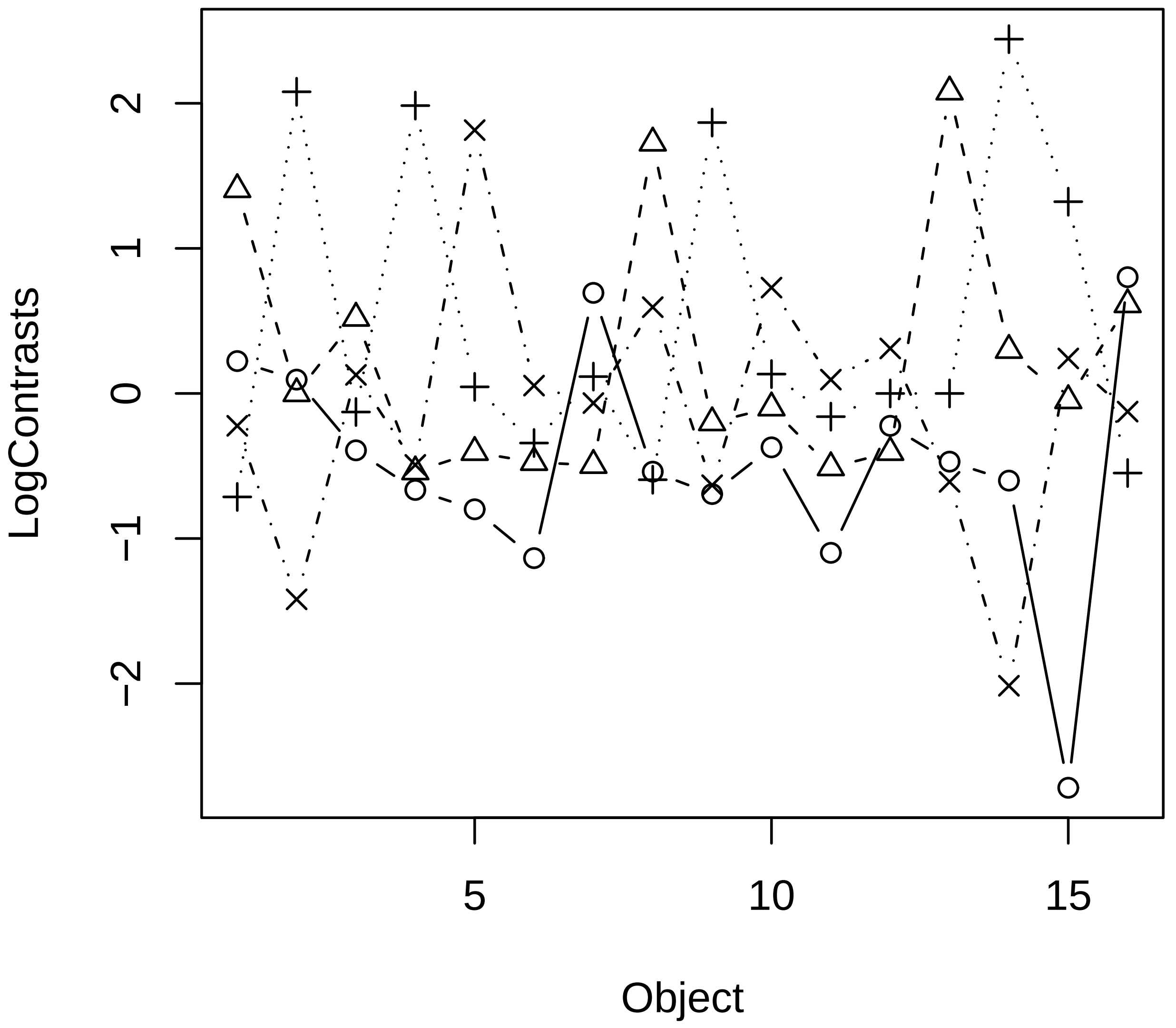
Some exploratory graphical analysis first:

```
> boxplot(as.data.frame(elelc))
> plot(as.ts(elelc),plot.t="single",type="b",pch=1:4,lty=1:4)
```

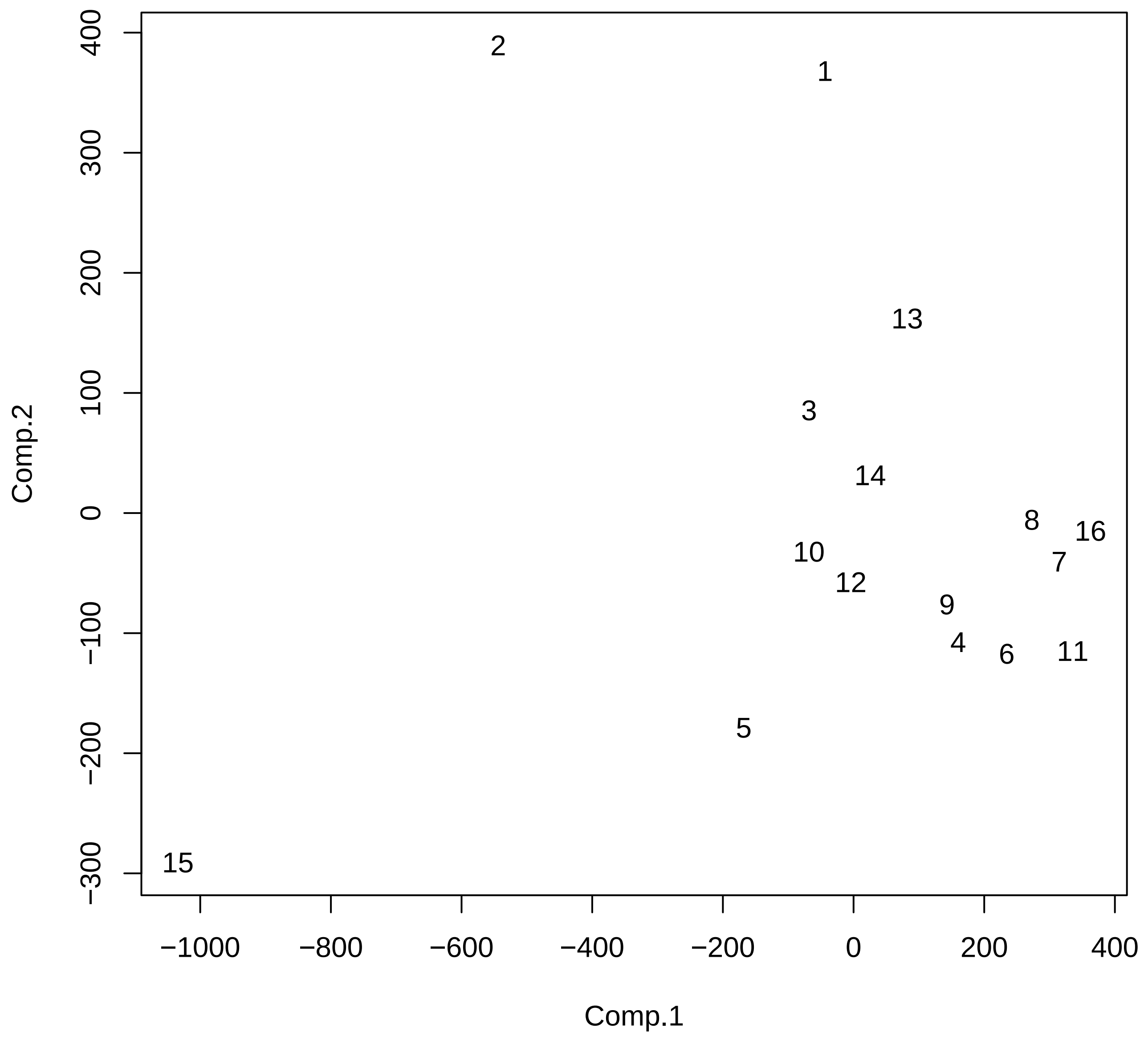
## Electrodes: boxplots of contrasts



## Contrasts: objectwise



## Contrasts: principal components



## Finally, we can look at Hotelling's $T^2$

```
> dim(elelc)
[1] 16  4
> elmu = apply(elelc,2,mean); elmu
[1] -0.4932261  0.2166766  0.4690089 -0.1009543
> elis = solve(var(elelc)) # oh: I am not supposed to do this!
> 1-pf(elmu %*% elis %*% elmu*16*12/(4*15),4,12)
0.1044359
```

First $p$-value here. Now, there is a suspicion that 15th subject is somewhat out of the ordinary (an "outlier", so to say) - how about trying the analysis without it?

```
> elmu=apply(elelc[-15,],2,mean)
> elis=solve(var(elelc[-15,]))
> 1-pf( elmu %*% elis %*% elmu * 15*11/(4*14),4,11)
0.1278355
```

How about looking only on the 1st and 3rd contrasts (together)?

```
> elmu = apply(elelc[,c(1,3)],2,mean)
> elis = solve(var(elelc[,c(1,3)]))
> 1-pf(elmu %*% elis %*% elmu*16*14/(2*15),2,14)
0.06439189
```

## $T^2$ continued

And now, look at the 1st and 3rd contrasts separately

```
> elmu = mean(elelc[,1])
> elis = solve(var(elelc[,1]))
> 1-pf(elmu %*% elis %*% elmu*16*15/(1*15),1,15)
0.02852685
> elmu = mean(elelc[,3])
> elis = solve(var(elelc[,3]))
> 1-pf(elmu %*% elis %*% elmu*16*15/(1*15),1,15)
0.1010562
```

Yeah, so the 1st one would yield some significance - only, we could arrive to this by a simple paired t-test...

```
> 4*mean(elelc[,1])/sqrt(var(elelc[,1]))
-2.422740
> qt(.025,15)
-2.131450
> 2*(1-pt(abs(4*mean(elelc[,1]))/sqrt(var(elelc[,1])),15))
0.02852685
```

Isn't there a function for t-test in R?

## Surely is!

```
> t.test(elel[,1],elel[,2],paired=TRUE)

        Paired t-test

data:  elel[, 1] and elel[, 2]
t = -2.4227, df = 15, p-value = 0.02853
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.92715075 -0.05930155
sample estimates:
mean of the differences
             -0.4932261
```

## Isn't there any package for Hotelling?

```
> summary(manova(elelc~1),intercept=T,test="H")
            Df Hotelling-Lawley approx F num Df den Df Pr(>F)
(Intercept) 1           0.81125  2.43374      4     12 0.1044
Residuals   15
> summary(manova(elelc~1),intercept=T,test="W")
            Df   Wilks approx F num Df den Df Pr(>F)
(Intercept) 1 0.55211  2.43374      4     12 0.1044
Residuals   15
> summary(manova(elelc[-15,]~1),intercept=T,test="H")
            Df Hotelling-Lawley approx F num Df den Df Pr(>F)
(Intercept) 1           0.82388  2.26568      4     11 0.1278
Residuals   14
> summary(manova(elelc[-15,]~1),intercept=T,test="R")
            Df   Wilks approx F num Df den Df Pr(>F)
(Intercept) 1 0.54828  2.26568      4     11 0.1278
Residuals   14
> summary(manova(elelc[-15,]~1),intercept=T,test="P")
            Df  Pillai approx F num Df den Df Pr(>F)
(Intercept)  1 0.45172   2.2657      4     11 0.1278
Residuals   14
```

## Hotelling package continued

```
> summary(manova(elelc[,c(1,3)]~1),intercept=T)
            Df  Pillai approx F num Df den Df  Pr(>F)
(Intercept)  1 0.32418   3.3577      2     14 0.06439 .
...

> summary(manova(elelc[,1]~1),intercept=T)
Error in manova(elelc[, 1] ~ 1) : need multiple response
> summary(aov(elelc[,1]~1),intercept=T)
            Df Sum Sq Mean Sq F value Pr(>F)
(Intercept)  1  3.892   3.892    5.87 0.0285 *
...
```

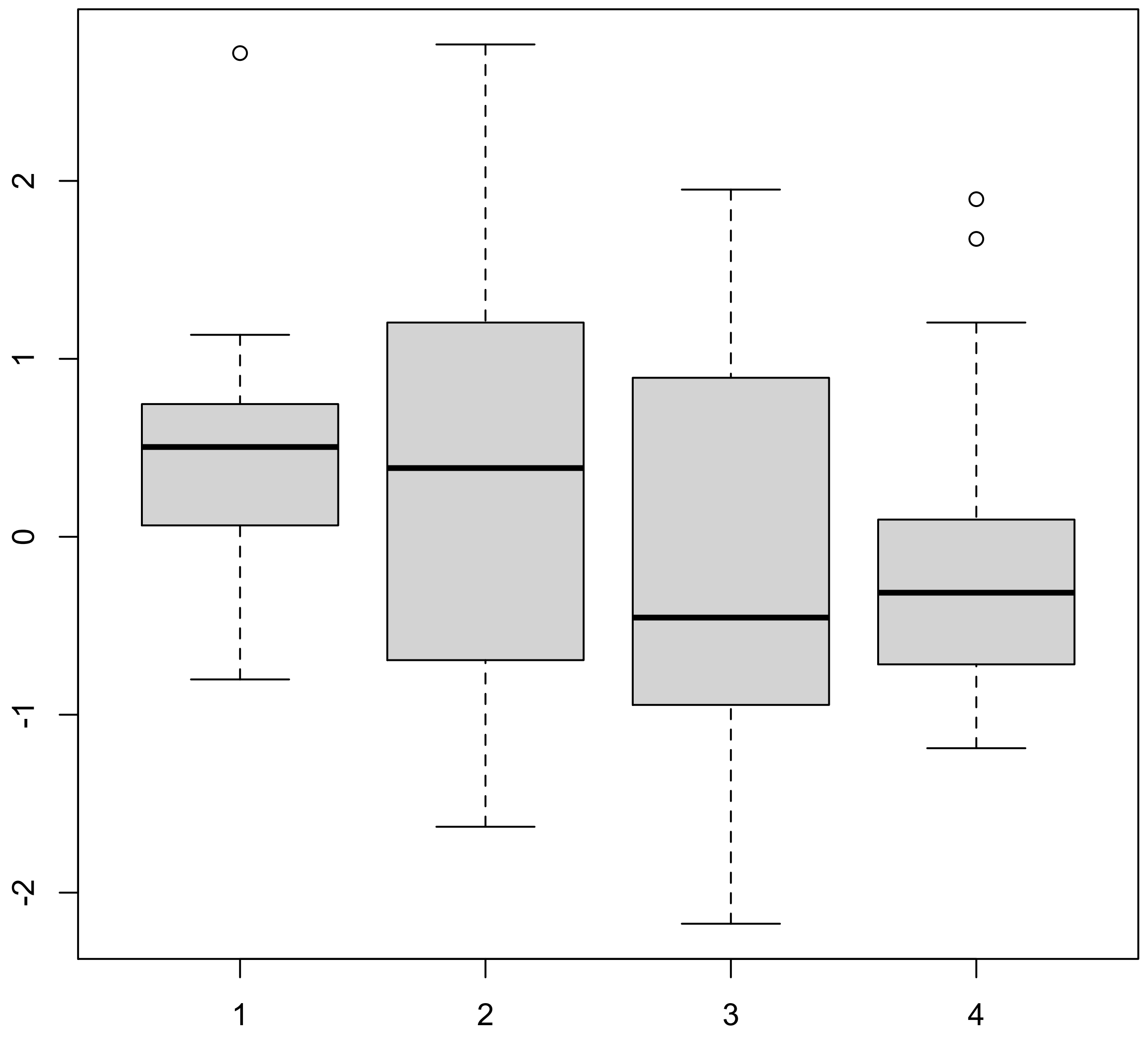
Recall:

```
> t.test(elel[,1],elel[,2],paired=TRUE)

Paired t-test

data:  elel[, 1] and elel[, 2]
t = -2.4227, df = 15, p-value = 0.02853
...
```

## Second thoughts: how about trying some other contrast matrix?

```
> contr.sum(5)[c(5,1:4),]
  [,1] [,2] [,3] [,4]
5   -1   -1   -1   -1
1    1    0    0    0
2    0    1    0    0
3    0    0    1    0
4    0    0    0    1
> elc=as.matrix(elel) %*% contr.sum(5)[c(5,1:4),]
> boxplot(elc)
```

## Boxplots of contrasts: not that bad



## Oops... was this expected?

```
> summary(manova(elc~1),intercept=T)
            Df  Pillai approx F num Df den Df Pr(>F)
(Intercept)  1 0.44789   2.4337      4     12 0.1044
Residuals   15
```

Well, was to be expected... or not? Also

```
> summary(manova(elc[-15,]~1),intercept=T)
            Df  Pillai approx F num Df den Df Pr(>F)
(Intercept)  1 0.45172   2.2657      4     11 0.1278
Residuals   14
```

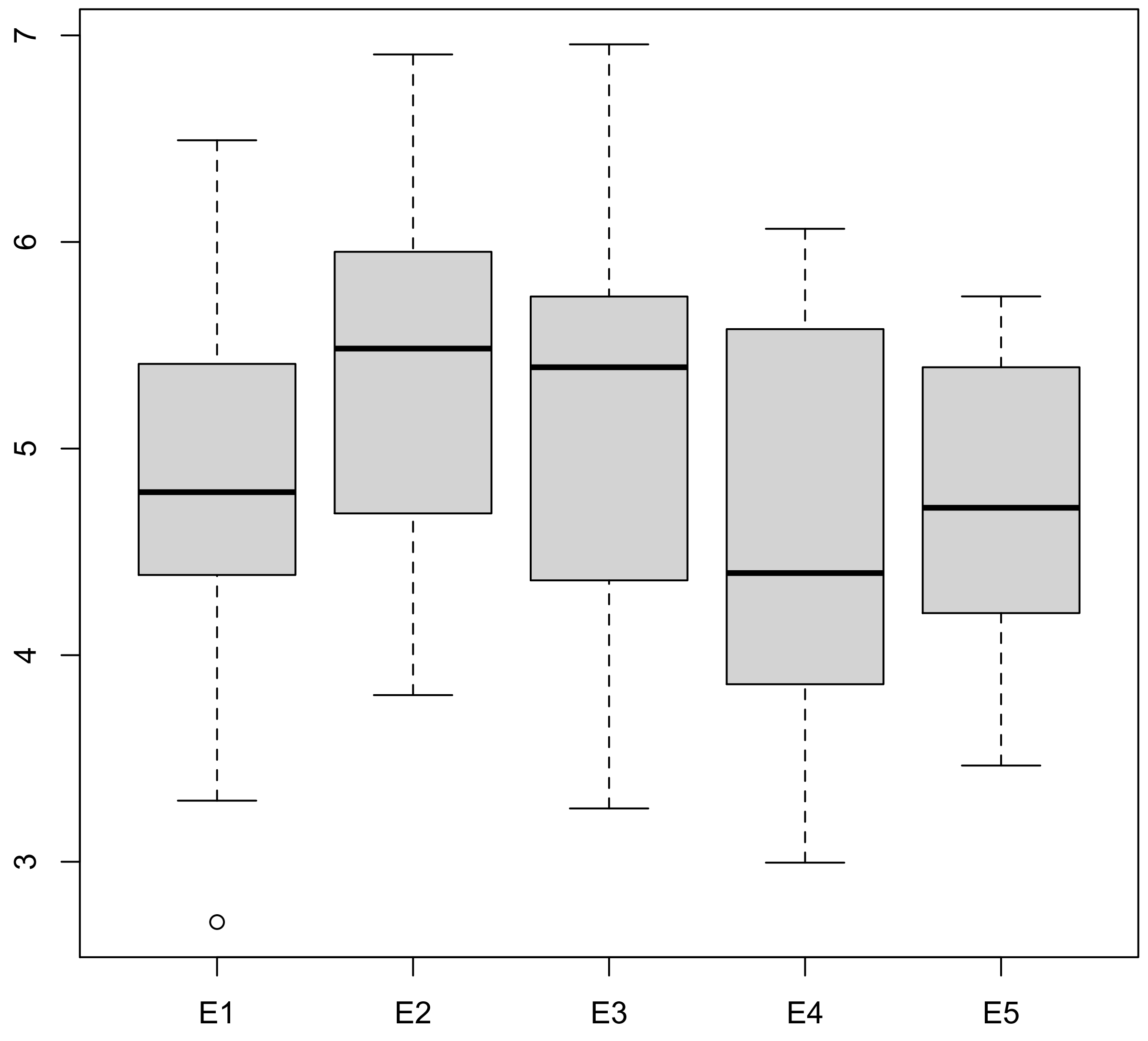
## For subsets, however...

```
> summary(manova(elc[,c(1,2,3)]~1),intercept=T)
            Df  Pillai approx F num Df den Df  Pr(>F)
(Intercept)  1 0.42713   3.2309      3     13 0.05757 .
...
> summary(manova(elc[,c(1,2)]~1),intercept=T)
            Df Pillai approx F num Df den Df Pr(>F)
(Intercept)  1 0.3549    3.851      2     14 0.0465 *
...
> summary(manova(elc[,c(1,3)]~1),intercept=T)
            Df Pillai approx F num Df den Df  Pr(>F)
(Intercept)  1 0.4194   5.0564      2     14 0.02224 *
...
```

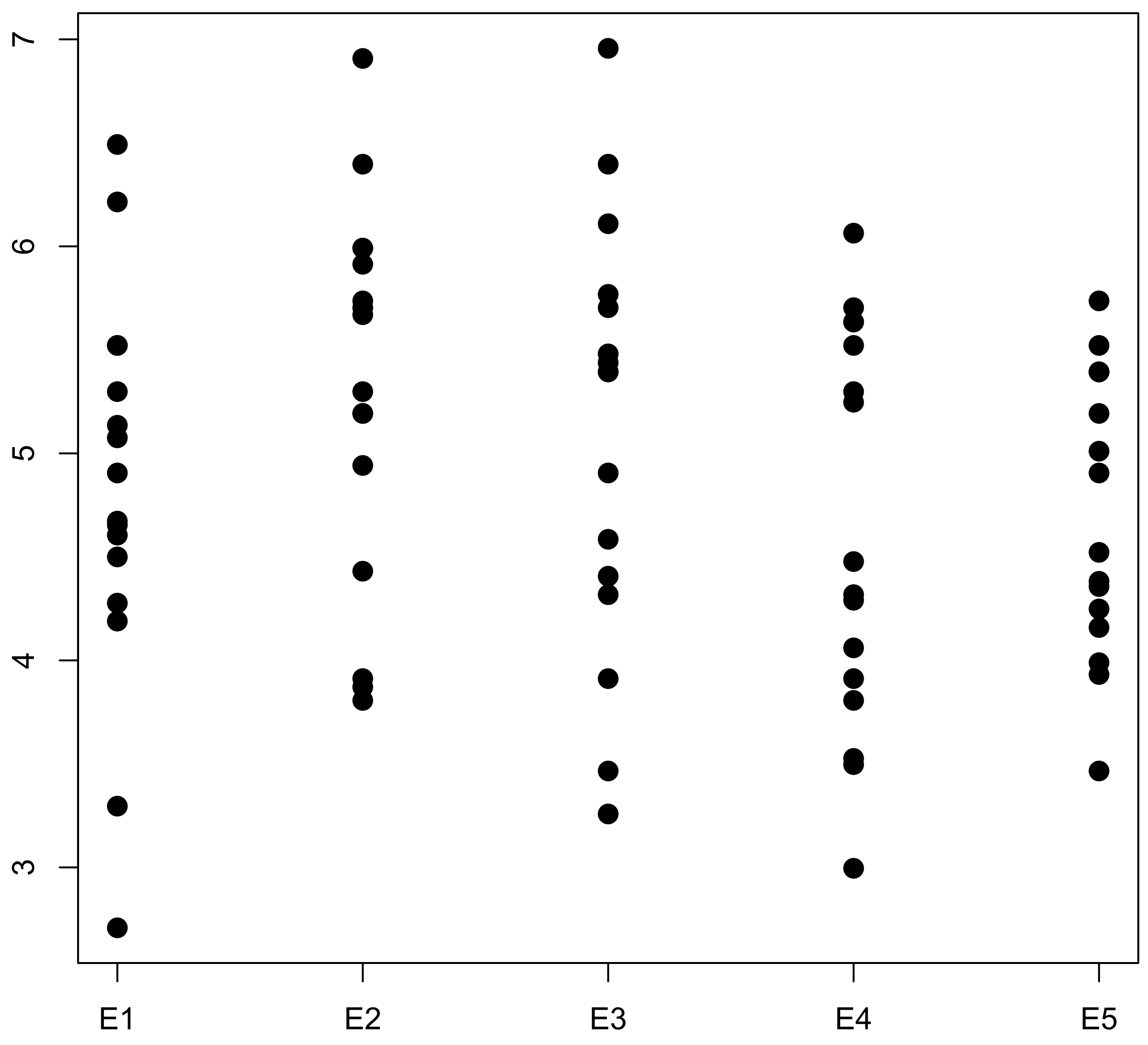
Note the “progress”: originally we have been only able to isolate one contrast significant at $\alpha = 0.05$. Now we are able to pick up two...

Shall we continue? (Better not, but I did...)

## Incidentally, the boxplots of original logarithms



## And something better than boxplots here(size!)



## A fancy contrast matrix?

Some impasses are inevitable:

```
> cn = rbind(c(-1,-1,0,0),c(1,0,-1,0),c(0,0,0,-1),
+ c(0,0,0,1),c(0,1,1,0))
> cn
     [,1] [,2] [,3] [,4]
[1,]   -1   -1    0    0
[2,]    1    0   -1    0
[3,]    0    0    0   -1
[4,]    0    0    0    1
[5,]    0    1    1    0
> ecn=as.matrix(elel) %*% cn
> summary(manova(ecn[,c(1,2,3)]~1),intercept=T)
Error in summary.manova(manova(ecc[, c(1, 2, 3)] ~ 1), intercept = T
  residuals have rank 2 < 3
> summary(manova(ecn[,c(1,3)]~1),intercept=T)
            Df  Pillai approx F num Df den Df  Pr(>F)
(Intercept)  1 0.42668   5.2097      2     14 0.02036 *
...
```

I told you: it has to have independent lines - in this case, columns

## Pairwise t-tests? Bonferroni? Holm?

```
> t.test(elel$E2,elel$E5,paired=T)$p.value
[1] 0.009286861
> t.test(elel$E2,elel$E4,paired=T)$p.value
[1] 0.02635447
> t.test(elel$E1,elel$E2,paired=T)$p.value
[1] 0.02852685
> t.test(elel$E3,elel$E4,paired=T)$p.value
[1] 0.1010562
> t.test(elel$E3,elel$E5,paired=T)$p.value
[1] 0.1023432
> t.test(elel$E2,elel$E3,paired=T)$p.value
[1] 0.3204591
> t.test(elel$E1,elel$E3,paired=T)$p.value
[1] 0.4026263
> t.test(elel$E1,elel$E4,paired=T)$p.value
[1] 0.5657955
> t.test(elel$E4,elel$E5,paired=T)$p.value
[1] 0.6492741
> t.test(elel$E1,elel$E5,paired=T)$p.value
[1] 0.6964895
```

## Maybe just a few but selected p-values

```
> summary(manova(elc[,c(1,2,3)]~1),intercept=T) # 5 out
            Df  Pillai approx F num Df den Df  Pr(>F)
(Intercept)  1 0.42713   3.2309      3     13 0.05757 .
...
> summary(manova(elc[,c(1,2,4)]~1),intercept=T) # 4 out
            Df  Pillai approx F num Df den Df  Pr(>F)
(Intercept)  1 0.42832   3.2467      3     13 0.05686 .
...
> summary(manova(elc[,c(1,3,4)]~1),intercept=T) # 3 out
            Df  Pillai approx F num Df den Df  Pr(>F)
(Intercept)  1 0.44756   3.5106      3     13 0.04628 *
...
> summary(manova(elc[,c(2,3,4)]~1),intercept=T) # 2 out
            Df  Pillai approx F num Df den Df Pr(>F)
(Intercept)  1 0.20921   1.1464      3     13 0.3672
...
```

(I like the last one, in a sense)

## So what?

And 1 out? Only possible with the other contrast matrix...

```
> summary(manova(elelc[,c(2,3,4)]~1),intercept=T)
            Df  Pillai approx F num Df den Df  Pr(>F)
(Intercept)  1 0.39456    2.824      3     13 0.08009 .
...
```

Ok, if we single 2 out, we are left with 1,3,4,5. The 3 looks quite spread, it may spoil the significance

```
> summary(manova(elc[,c(3,4)]~1),intercept=T)
            Df   Pillai approx F num Df den Df Pr(>F)
(Intercept)  1 0.022733  0.16283      2     14 0.8513
...
```

Seems like 1,4,5 are quite close... So how about abandoning 3 (see t-tests), keeping 2, and out of 1,4,5 keep 5 (see t-tests)

## Like this?

```
> t.test(elel$E2,elel$E5,paired=T)$p.value
[1] 0.009286861
> t.test(elel$E2,elel$E4,paired=T)$p.value
[1] 0.02635447
> t.test(elel$E1,elel$E2,paired=T)$p.value
[1] 0.02852685
> t.test(elel$E2,elel$E3,paired=T)$p.value
[1] 0.3204591
> summary(manova(elc[,c(2,3,4)]~1),intercept=T) # 2 out
            Df  Pillai approx F num Df den Df Pr(>F)
(Intercept)  1 0.26456   1.1464      3     13 0.3672
...
> summary(manova(elc[,c(3,4)]~1),intercept=T)
            Df   Pillai approx F num Df den Df Pr(>F)
(Intercept)  1 0.022733  0.16283      2     14 0.8513
...
```

Oops... $0.05/6 = 0.008333$

$0.05/5 = 0.01 > 0.009$

## Omit one!

```
> t.test(elel$E2,elel$E5,paired=T)$p.value
[1] 0.009286861
> t.test(elel$E2,elel$E4,paired=T)$p.value
[1] 0.02635447
> t.test(elel$E1,elel$E2,paired=T)$p.value
[1] 0.02852685
> summary(manova(elc[,c(2,3,4)]~1),intercept=T) # 2 out
            Df  Pillai approx F num Df den Df Pr(>F)
(Intercept)  1 0.20921   1.1464      3     13 0.3672
...
> summary(manova(elc[,c(3,4)]~1),intercept=T)
            Df   Pillai approx F num Df den Df Pr(>F)
(Intercept)  1 0.022733  0.16283      2     14 0.8513
...
```

## Or maybe better

```
> t.test(elel$E2,elel$E5,paired=T)$p.value
[1] 0.009286861
> t.test(elel$E2,elel$E4,paired=T)$p.value
[1] 0.02635447
> t.test(elel$E1,elel$E2,paired=T)$p.value
[1] 0.02852685
> summary(manova(elc[,c(2,3,4)]~1),intercept=T) # 2 out
            Df  Pillai approx F num Df den Df Pr(>F)
(Intercept)  1 0.20921   1.1464      3     13 0.3672
...
> summary(manova(elc[,c(3,4)]~1),intercept=T)
            Df   Pillai approx F num Df den Df Pr(>F)
(Intercept)  1 0.022733  0.16283      2     14 0.8513
...
```

That is, keep 2 and 5 (perhaps...)

## More Extending Classics: Two-Sample Hoteling's $T^2$

Postponed past MANOVA

## Trying to Extend Classics: Linear Model and ANOVA

## It starts easy: estimation

As in the univariate case, we are going to use least squares, which tare not only motivated by numerical convenience (quadratic minimization problem leads to linear equations), Gauss-Markov theory (best linear unbiased estimate), and result as maximum likelihood estimates under normality assumption,

but have also a particular advantage in the multivariate context: estimation of a multivariate model splits into the estimation of $p$ univariate ones - minimizing the left-hand sum amounts to separate minimizations of the right-hand sums in

$$\sum_{i,j}(y_{ij}-\mathbf{x}_i^{\mathsf{T}}\boldsymbol{\beta}_j)^2=\sum_i(y_{i1}-\mathbf{x}_i^{\mathsf{T}}\boldsymbol{\beta}_1)^2+\cdots+\sum_i(y_{ip}-\mathbf{x}_i^{\mathsf{T}}\boldsymbol{\beta}_p)^2$$

This is not true in general for other univariate regression estimators

## In matrix language it looks almost the same

First, $\mathbf{X}$ is an $n \times q$ matrix of covariates, same for the all $p$ responses $\mathbf{y}_j$, $j = 1, 2, \ldots, p$; we have $p$ linear models

$$\mathbf{y}_j = \mathbf{X}\boldsymbol{\beta}_j + \boldsymbol{\varepsilon}_j$$

Now, we glue columnwise all $p$

responses $\mathbf{y}_j$ into an $n \times p$ matrix $\mathbf{Y}$

parameter vectors $\boldsymbol{\beta}_j$ into an $q \times p$ matrix $\mathbf{B}$

$\boldsymbol{\varepsilon}_j$ vectors into one $n \times p$ matrix $\mathbf{E}$

to obtain the formulation of the multivariate linear model

$$\mathbf{Y} = \mathbf{X}\mathrm{B} + \mathbf{E}$$

Due to the aforementioned split of the least-squares criterion, the $p$ prescriptions for estimates in the $p$ linear models

$$\widehat{\boldsymbol{\beta}}_j = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}_j$$

can be thus aggregated into one

$$\widehat{\mathbf{B}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y}$$

## Componentwise

$y_{ij} = \mathbf{x}_{i\cdot}^\top \boldsymbol{\beta}_{\cdot j} + \varepsilon_{ij},\ i = 1, \ldots, n,\ j = 1, \ldots, p \qquad \mathbf{X} = \begin{pmatrix} \mathbf{x}_{1\cdot}^\top \\ \vdots \\ \mathbf{x}_{i\cdot}^\top \\ \vdots \\ \mathbf{x}_{n\cdot}^\top \end{pmatrix}$

Typical modeling assumptions: $\varepsilon_i$ are now **rows** of $\mathbf{E} = \begin{pmatrix} \boldsymbol{\varepsilon}_{1\cdot}^\top \\ \vdots \\ \boldsymbol{\varepsilon}_{i\cdot}^\top \\ \vdots \\ \boldsymbol{\varepsilon}_{n\cdot}^\top \end{pmatrix}$

$\mathrm{E}(\boldsymbol{\varepsilon}_{i\cdot}) = \mathbf{0}$

$\mathrm{Var}(\boldsymbol{\varepsilon}_{i\cdot}) = \boldsymbol{\Sigma}$ (not depending on $j$)

$\mathrm{Cov}(\boldsymbol{\varepsilon}_{i\cdot}, \boldsymbol{\varepsilon}_{k\cdot}) = \mathbf{0}$ for $i \neq k$, or stronger: rows $\boldsymbol{\varepsilon}_{i\cdot}$ are independent

also normality assumption: $\boldsymbol{\varepsilon}_{i\cdot}$ are independent, each $N(\mathbf{0}, \boldsymbol{\Sigma})$

## Verifying the maximum likelihood claim

If we assume a **normal multivariate linear model**, in which all lines of the matrix $\mathbf{E}$ arise via iid sampling with the common distribution being normal, $N(\mathbf{0}, \mathbf{\Sigma})$, wioth positive definite (nonsingular) $\mathbf{\Sigma}$ (so that its density exists). we can write the negative of the log-likelihood is, up to an additive and multiplicative constants that don't affect the outcomes of minimization, as

$$\sum_i \log\det(\mathbf{\Sigma}) + (\mathbf{y}_{i\cdot} - \mathbf{x}_{i\cdot}^\top \mathbf{B})\mathbf{\Sigma}^{-1}(\mathbf{y}_{i\cdot} - \mathbf{x}_{i\cdot}^\top \mathbf{B})^\top$$

$$= n\log\det(\mathbf{\Sigma}) + \sum_i \operatorname{tr}\Big((\mathbf{y}_{i\cdot} - \mathbf{x}_{i\cdot}^\top \mathbf{B})\mathbf{\Sigma}^{-1}(\mathbf{y}_{i\cdot} - \mathbf{x}_{i\cdot}^\top \mathbf{B})^\top\Big)$$

$$= n\log\det(\mathbf{\Sigma}) + \operatorname{tr}\Big(\sum_i (\mathbf{y}_{i\cdot} - \mathbf{x}_{i\cdot}^\top \mathbf{B})^\top(\mathbf{y}_{i\cdot} - \mathbf{x}_{i\cdot}^\top \mathbf{B})\mathbf{\Sigma}^{-1}\Big)$$

$$= n\log\det(\mathbf{\Sigma}) + \operatorname{tr}\Big((\mathbf{Y} - \mathbf{X}\mathbf{B})^\top(\mathbf{Y} - \mathbf{X}\mathbf{B})\mathbf{\Sigma}^{-1}\Big)$$

If $\mathbf{\Sigma}$ is fixed, then the first term does not depend on $\mathbf{B}$ and the second one is minimized by $\mathbf{B}$ equal to the least squares estimator $\widehat{\mathbf{B}}$:

## Verifying continued

$$\operatorname{tr}\Big((\mathbf{Y}-\mathbf{X}\mathbf{B})^\top(\mathbf{Y}-\mathbf{X}\mathbf{B})\mathbf{\Sigma}^{-1}\Big)$$

$$= \operatorname{tr}\Big((\mathbf{Y}-\mathbf{X}\widehat{\mathbf{B}})^\top(\mathbf{Y}-\mathbf{X}\widehat{\mathbf{B}})\mathbf{\Sigma}^{-1}\Big) + \operatorname{tr}\Big((\mathbf{X}\widehat{\mathbf{B}}-\mathbf{X}\mathbf{B})^\top(\mathbf{X}\widehat{\mathbf{B}}-\mathbf{X}\mathbf{B})\mathbf{\Sigma}^{-1}\Big)$$

$$= \operatorname{tr}\Big((\mathbf{Y}-\mathbf{X}\widehat{\mathbf{B}})^\top(\mathbf{Y}-\mathbf{X}\widehat{\mathbf{B}})\mathbf{\Sigma}^{-1}\Big) + \operatorname{tr}\Big((\mathbf{X}\widehat{\mathbf{B}}-\mathbf{X}\mathbf{B})\mathbf{\Sigma}^{-1}(\mathbf{X}\widehat{\mathbf{B}}-\mathbf{X}\mathbf{B})^\top\Big)$$

and the last term is the trace of the nonnegative definite matrix.

Put $\mathbf{B} = \widehat{\mathbf{B}}$; let $\mathbf{A} = (\mathbf{Y}-\mathbf{X}\widehat{\mathbf{B}})^\top(\mathbf{Y}-\mathbf{X}\widehat{\mathbf{B}})$

the negative log-likelihood becomes, if $\mathbf{A}$ is nonsingular,

$n \log\det(\mathbf{\Sigma}) + \operatorname{tr}(\mathbf{A}\mathbf{\Sigma}^{-1}) = -n\log\det(\mathbf{\Sigma}^{-1}) + \operatorname{tr}(\mathbf{A}\mathbf{\Sigma}^{-1})$

$= -n\log\det(\mathbf{A}\mathbf{\Sigma}^{-1}) + n\log\det\mathbf{A} + \operatorname{tr}(\mathbf{A}\mathbf{\Sigma}^{-1})$

which is minimized by $\hat{\mathbf{\Sigma}} = \dfrac{1}{n}\mathbf{A}$ if $\mathbf{A}$ is fixed (and nonsingular)

## Least squares: in X pretty much the same

We have $\mathbf{X}$, which generates the model $\mathcal{M}(\mathbf{X}) = \mathrm{Im}(\mathbf{X})$

with dimension typically equal to the rank $p$ of $\mathbf{X}$

Then $\mathbf{X}_0$ generates a *submodel* $\mathcal{M}(\mathbf{X}_0) = \mathrm{Im}(\mathbf{X}_0) \subset \mathrm{Im}(\mathbf{X})$

with dimension, say, $p_0$

Least squares estimates yield corresponding projections:

$\mathbf{X}\widehat{\mathbf{B}}$ onto $\mathrm{Im}(\mathbf{X})$ and $\mathbf{X}\widehat{\mathbf{B}}_0$ onto $\mathrm{Im}(\mathbf{X}_0)$

We obtain the residuals, which then belong to the orthogonal complements:

$\mathbf{Y} - \mathbf{X}\widehat{\mathbf{B}}$ to $\mathrm{Im}(\mathbf{X})^{\perp}$ with dimension $n - p$

traditionally referred to as df (degrees of freedom)

$\mathbf{Y} - \mathbf{X}\widehat{\mathbf{B}}_0$ to $\mathrm{Im}(\mathbf{X}_0)^{\perp}$ with dimension $n - p_0$

## The ANOVA decomposition

Then $\mathbf{X}\hat{\mathbf{B}} - \mathbf{X}\hat{\mathbf{B}}_0$ belongs to $\mathrm{Im}(\mathbf{X}) \cap \mathrm{Im}(\mathbf{X}_0)^\perp$ which has dimension $p - p_0$, which is traditionally referred to as degrees of freedom of the submodel (null hypothesis) $\mathrm{df}_{\mathrm{H}}$; it is the rank of $\mathbf{X}\hat{\mathbf{B}} - \mathbf{X}\hat{\mathbf{B}}_0$

Therefore, any column of $\mathbf{X}\hat{\mathbf{B}} - \mathbf{X}\hat{\mathbf{B}}_0$ is orthogonal to any column of $\mathbf{Y} - \mathbf{X}\hat{\mathbf{B}}$ - the fact that yields...

$$\begin{aligned}
\mathbf{T} &= (\mathbf{Y} - \mathbf{X}\hat{\mathbf{B}}_0)^\top(\mathbf{Y} - \mathbf{X}\hat{\mathbf{B}}_0) \\
&= (\mathbf{Y} - \mathbf{X}\hat{\mathbf{B}} + \mathbf{X}\hat{\mathbf{B}} - \mathbf{X}\hat{\mathbf{B}}_0)^\top(\mathbf{Y} - \mathbf{X}\hat{\mathbf{B}} + \mathbf{X}\hat{\mathbf{B}} - \mathbf{X}\hat{\mathbf{B}}_0) \\
&= (\mathbf{Y} - \mathbf{X}\hat{\mathbf{B}})^\top(\mathbf{Y} - \mathbf{X}\hat{\mathbf{B}}) + (\mathbf{X}\hat{\mathbf{B}} - \mathbf{X}\hat{\mathbf{B}}_0)^\top(\mathbf{X}\hat{\mathbf{B}} - \mathbf{X}\hat{\mathbf{B}}_0) \\
&= \mathbf{W} + (\mathbf{T} - \mathbf{W}) = \mathbf{W} + \mathbf{H}
\end{aligned}$$

by the orthogonality of columns of $\mathbf{Y} - \mathbf{X}\hat{\mathbf{B}}$ and $\mathbf{X}\hat{\mathbf{B}} - \mathbf{X}\hat{\mathbf{B}}_0$ - this also implies that under iid sampling model for the matrix $\mathbf{E}$ these matrices are uncorrelated, which under assumption of normal distribution further strengthens to independence and subsequently to the independence of $\mathbf{W}$ and $\mathbf{H}$

Note that the rank of $\mathbf{H}$ is equal to the rank of $\mathrm{df}_{\mathrm{H}}$, as the rank of $(\mathbf{X}\hat{\mathbf{B}} - \mathbf{X}\hat{\mathbf{B}}_0)^\top(\mathbf{X}\hat{\mathbf{B}} - \mathbf{X}\hat{\mathbf{B}}_0)$ is equal to the rank of $\mathbf{X}\hat{\mathbf{B}} - \mathbf{X}\hat{\mathbf{B}}_0$

## ANOVA decomposition: uni- and multivariate

In the univariate case, the elements of this decomposition are nonnegative numbers; now they are nonnegative definite matrices. Following the univariate analogy, we compare matrices "for size"

In univariate setting, we have the decomposition

$$RSS_H = RSS + (RSS_H - RSS)$$

containing nonnegative numbers - corresponding in our notation to

$$\mathbf{T} = \mathbf{W} + (\mathbf{T} - \mathbf{W}) = \mathbf{W} + \mathbf{H}$$

- the decomposition to nonnegative definite matrices

These represent minimized sum of squares, and serve also as a basis for variance-covariance estimation

Unlike in the univariate case, we cannot calculate a simple ratio $(RSS_H - RSS)/RSS$; even if we replace it by $\mathbf{H}\mathbf{W}^{-1}$ (or $\mathbf{W}^{-1}\mathbf{H}$), we face the problem of evaluating "the size" of the resulting $p \times p$ matrix

## First step: discriminants (canonical variates)

The natural first step to such an evaluation are the eigenvalues of $\mathbf{HW}^{-1}$ (we know[8] those are the same, at least the nonzero ones, as the eigenvalues of $\mathbf{W}^{-1}\mathbf{H}$)

(It can be also shown, at least for some types of hypotheses, that if the tests have to be invariant with respect to origin and scale of the data, the test statistic must depend on these eigenvalues)

The eigenvalues of $\mathbf{HW}^{-1}$ appear in the solution of the optimization problem[9] that seeks $\mathbf{x}$ maximizing

$$\frac{\mathbf{x}^\top \mathbf{H}\mathbf{x}}{\mathbf{x}^\top \mathbf{W}\mathbf{x}}$$

Vectors $\mathbf{x}$ maximizing this expression form a direction: if $\mathbf{x}$ is a maximizer, then so is $c\mathbf{x}$, for any $c \neq 0$. If we project the data $\mathbf{Y}$ on the direction (picking up $\mathbf{x}$ with $\|\mathbf{x}\| = 1$), we obtain the first **linear discriminant**

[8] see Problem 1
[9] see Problem 9

## And so on

The direction of the second linear discriminant is obtained by the maximization of the same expression as above, only restricted to the directions orthogonal to the direction of the first discriminant

The direction of the third linear discriminant then maximizes the same expression, among directions orthogonal to the directions of the first and second one - and so on

The maximizing directions correspond to the eigenvalues of $\mathbf{HW}^{-1}$, the directions are the corresponding eigenvectors, and **linear discriminants** are projections of $\mathbf{Y}$ to these directions

And all this makes sense only when the maximization and ensuing projections are nontrivial: linear discriminants thus correspond *only* to the *nonzero eigenvalues* of $\mathbf{HW}^{-1}$ - their number is thus related to the number of nonzero eigenvalues (= rank) of $\mathbf{HW}^{-1}$

- which is $\mathrm{df}_{\mathrm{H}}$, the rank of $\mathbf{H}$, as that is less than df, the rank of $\mathbf{W}$

Linear discriminants are mostly known as *Fisher linear discriminants* in the specific setting of one-way layout

## Recall first: one-way layout univariate

The predicted variable is $y_{ki}$, comes in $g = 1, \dots, K$ groups

The model is $y_{gi} = \mu_g + \varepsilon_{gi}$ (or some equivalent form)

The sizes of groups are not necessarily equal: $n = n_1 + \cdots + n_K$
for each $g$, $i = 1, 2, \dots, n_g$

The errors $\varepsilon_{gi}$ are independent and their distribution is $N(0, \sigma^2)$
(the same variance for all errors!)

The hypothesis of interest is: all $\mu_g$ equal, $\mu_1 = \mu_2 = \cdots = \mu_K$

The predicted values via the least-squares method are:

under the model: $\bar{y}_g$, the mean in the group

under restricted model (hypothesis): $\bar{y}$, the mean of all $y_{gi}$

# One-way layout: univariate - testing

Submodel (restricted model): we fit the same $\mu$ to all $y_{gi}$
its estimate is grand mean $\bar{y}$, over all $y_{gi}$
the minimized sum of squares is $T = \sum(y_{gi} - \bar{y})^2$
divided by $n - 1$ degrees of freedom gives the estimate $s_T^2$

Full model: we fit not necessarily same $\mu_g$ to each group of $y_{gi}$
the estimates are group means $\bar{y}_g$
the minimized sum of squares is $W = \sum(y_{gi} - \bar{y}_g)^2$
divided by $n - K$ degrees of freedom gives $s^2 = s_W^2$

The ratio $\dfrac{\dfrac{T - W}{n - 1 - (n - K)}}{\dfrac{W}{n - K}} = \dfrac{\dfrac{T - W}{K - 1}}{\dfrac{W}{n - K}}$ is good for testing

if the assumptions hold, its distribution is $F_{(K-1),(n-K)}$
and if the hypothesis is true, it tends to be small

## One-way layout: multivariate

The response is $y_{gij}$, coming in $g = 1, \dots, K$ groups
and having $j = 1, \dots, p$ components: it is $p$-dimensional

Again, the sizes of groups are not necessarily equal:
$n = n_1 + \cdots + n_K$ and for each $g$, $i = 1, 2, \dots, n_g$

The model is $y_{gij} = \mu_{gj} + \varepsilon_{gij}$
$\boldsymbol{\mu}_g$ is a vector with components $\mu_{gj}$
$\boldsymbol{\varepsilon}_{gi}$ a random vector with components $\varepsilon_{gij}$
and $\mathbf{y}_{gi}$ a vector with components $y_{gij}$ - the transposed row of $\mathbf{Y}$, as all vectors are $p$-dimensional *column* vectors

The errors $\boldsymbol{\varepsilon}_{gi}$ are independent and have distribution $N(\mathbf{0},\Sigma)$
(the same variance-covariance matrix for all errors!)

The hypothesis of interest: all $\boldsymbol{\mu}_g$ equal

Prediction from via the least-squares fit
under model, and
under submodel (restricted model, hypothesis)

## One-way layout: multivariate - testing

Submodel (restricted model): we fit the same $\boldsymbol{\mu}$ to all $\mathbf{y}_{gi}$

its estimate is the grand columnwise of $\mathbf{Y}$ mean $\bar{\mathbf{y}} = (\bar{y}_1, \dots, \bar{y}_p)^\top$

the estimate of variance-covariance matrix is obtained from

the matrix $\mathbf{T}$ with elements $T_{rs} = \sum_{g,i} (y_{gir} - \bar{y}_r)(y_{gis} - \bar{y}_s)$

$$\mathbf{T} = \sum_g \sum_{i=1}^{n_g} (\mathbf{y}_{gi} - \bar{\mathbf{y}})(\mathbf{y}_{gi} - \bar{\mathbf{y}})^\top$$

The full model: each $\mu_k$ in each group of $y_{ki}$ is fit separately

the estimates are group means $\bar{\mathbf{y}}_g = (\bar{y}_{g1}, \dots, \bar{y}_{gp})^\top$

the estimate of variance-covariance matrix is obtained from

the matrix $\mathbf{W}$ with elements $W_{rs} = \sum_{g,i} (y_{gir} - \bar{y}_{gr})(y_{gis} - \bar{y}_{gs})$

$$\mathbf{W} = \sum_g \sum_{i=1}^{n_g} (\mathbf{y}_{gi} - \mathbf{y}_g)(\mathbf{y}_{gi} - \mathbf{y}_g)^\top$$

Finally, the matrix $\mathbf{H} = \mathbf{T} - \mathbf{W} = \sum_g n_g (\bar{\mathbf{y}}_g - \bar{\mathbf{y}})(\bar{\mathbf{y}}_g - \bar{\mathbf{y}})^\top$

Note: dividing by degrees of freedom is omitted, as the distribution theory now does not require it

## So, how to evaluate the size by one number?

Among various possibilities that have been considered, people eventually settled for the following four

1. Hotelling's–Lawley's trace (a direct generalization of F ratio):
$$\operatorname{tr}\left(\mathbf{H}\mathbf{W}^{-1}\right) = \operatorname{tr}\left((\mathbf{T}-\mathbf{W})\mathbf{W}^{-1}\right)$$
2. Roy's maximal eigenvalue: of $\mathbf{H}\mathbf{W}^{-1} = (\mathbf{T}-\mathbf{W})\mathbf{W}^{-1}$
3. Pillai's (or Pillai's-Bartlett's) trace:
$$\begin{aligned}\operatorname{tr}\left(\mathbf{H}\mathbf{T}^{-1}\right) &= \operatorname{tr}\left((\mathbf{T}-\mathbf{W})\mathbf{T}^{-1}\right) = \operatorname{tr}\left(\mathbf{H}(\mathbf{W}+\mathbf{H})^{-1}\right)\\ &= \operatorname{tr}\left(\mathbf{H}\mathbf{W}^{-1}(\mathbf{I}+\mathbf{H}\mathbf{W}^{-1})^{-1}\right)\end{aligned}$$
4. Wilks's
$$\lambda = \frac{\det \mathbf{W}}{\det \mathbf{T}} = \frac{\det \mathbf{W}}{\det(\mathbf{W}+\mathbf{H})} = \det\left((\mathbf{I}+\mathbf{H}\mathbf{W}^{-1})^{-1}\right)$$

As stipulated, all of them are functions of the eigenvalues of $\mathbf{H}\mathbf{W}^{-1}$. Recall that $\mathbf{AB}$ and $\mathbf{BA}$ have the same nonzero eigenvalues[10]; in particular, maximal eigenvalues of $\mathbf{AB}$ and $\mathbf{BA}$ are equal; the eigenvalues of the inverse matrix are the reciprocals of the original ones; the eigenvalues of $\mathbf{I}+\mathbf{A}$ are $1+$ the eigenvalues of $\mathbf{A}$

[10] Problem 1

## Therefore

$\operatorname{tr}\mathbf{AB} = \operatorname{tr}\mathbf{BA}$; also, trace is the sum of eigenvalues

Determinant is the product of eigenvalues

Let $\lambda_1 \geqslant \lambda_2 \geqslant \ldots$ be eigenvalues of $\mathbf{HW}^{-1} = (\mathbf{T}-\mathbf{W})\mathbf{W}^{-1} = \mathbf{TW}^{-1}-\mathbf{I}$

Let $\xi_1 \geqslant \xi_2 \geqslant \ldots$ be eigenvalues of

$$\mathbf{HT}^{-1} = \mathbf{T}^{-1}\mathbf{H} = \mathbf{T}^{-1}(\mathbf{T}-\mathbf{W}) = \mathbf{I}-\mathbf{T}^{-1}\mathbf{W}$$

We have that $\xi_i = \dfrac{\lambda_i}{1+\lambda_i}$, $\lambda_i = \dfrac{\xi_i}{1-\xi_i}$

1. $\operatorname{tr}\left(\mathbf{HW}^{-1}\right) = \sum_i \lambda_i = \sum_i \dfrac{\xi_i}{1-\xi_i}$
2. The maximal eigenvalue of $\mathbf{HW}^{-1}$ is $\lambda_1 = \dfrac{\xi_1}{1-\xi_1}$
3. $\operatorname{tr}\left(\mathbf{HT}^{-1}\right) = \sum_i \dfrac{\lambda_i}{1+\lambda_i} = \sum_i \xi_i$
4. $\dfrac{\det(\mathbf{W})}{\det(\mathbf{T})} = \prod_i \dfrac{1}{1+\lambda_i} = \prod_i (1-\xi_i)$

# Apocrypha about distributions

Notation:

$\nu$: "residual degrees of freedom": $n - \dim \mathcal{M}(\mathbf{X})$
$\nu_H$: "hypothesis degrees of freedom": $\dim \mathcal{M}(\mathbf{X}) - \dim \mathcal{M}(\mathbf{X}_H)$
$p$: the dimension of the response
$t = \min\{p, \nu_H\}$, $u = \max\{p, \nu_H\}$, $\lambda = \frac{1}{4}(p\nu_H - 2)$,
$r = \frac{1}{2}(|p - \nu_H| - 1)$, $k = \nu - p - 1$, $m = \nu + \frac{1}{2}\nu_H - \frac{1}{2}(p+1)$,

There we quite a few approximations proposed (the following are not the only ones)

1. The distribution of Roy's maximal eigenvalue (root)

$$\Theta \qquad \text{which is} \qquad \frac{u}{\nu - u - 1} F_{u,\nu-u-1}$$

*gives only a lower bound on the approximate p-value*

(hence its significance is often ignored - especially when other criteria contraindicate it)

2. Hotelling's-Lawley's trace $U$

$$\frac{(tk+2)}{t^2(2r+t+1)} U \qquad \text{is approximately} \qquad F_{t(2r+t+1),tk+2}$$

## More apocrypha

3. Pillai's trace $V$:

$$\frac{k+t+1}{2r+t+1}\,\frac{V}{t-V} \qquad \text{is approximately} \qquad F_{t(2r+t+1),t(k+t+1)}$$

4. Wilks's $\Lambda$:

Bartlett's approximation: $-m \log \Lambda$ is approximately $\chi^2_{p\nu_H}$

Rao's transformation: $\dfrac{1-\Lambda^{1/s}}{\Lambda^{1/s}}\dfrac{ms-2\lambda}{p\nu_H}$ is

approximately (exactly if $t \leqslant 2$) $F_{p\nu_H, ms-2\lambda}$

non-integer degrees of freedom possible here

$s = \sqrt{\frac{p^2\nu_H^2-4}{p^2+\nu_H^2-5}}$ if $p^2+\nu_H^2-5>0$, otherwise $s=1$

All of the alternatives are approximated by F for certain constant multiples, as listed above; some of the alternatives feature also some other approximations, or even "exact" distributions (exact under exact normality).

## Distributions: summary (remember this!)

Once again: *only lower bound on p-value for Roy; hence generally disregarded if others are not significant*

Wilks's $F$ approximation may have non-integer (second) degrees of freedom; it has also standard approximation, by $\chi^2$, as a likelihood ratio - and also an exact distribution can be handled numerically

All four alternatives *approximately equivalent for large sample sizes*

All four are *equivalent exactly*

*for* $p = 1$*, that is, in the univariate case*
*when they reduce to the usual F-ratio*

*and also when df*$_{\mathrm{H}} = 1$ *- in particular for*

*one-sample or two-sample models - then equivalent to*
*one-sample or two-sample Hotelling* $T^2$*, respectively*

Some recommend as the first choice Pillai's trace $V$, because of its robustness; then Wilks's $\Lambda$; then Hotelling's-Lawley's trace $U$

## Likelihood ratio motivation for Wilks's $\Lambda$

Under normal distribution of errors

the maximized value of the likelihood is $\dfrac{e^{-np/2}}{(2\pi)^{np/2}(\det\widehat{\boldsymbol{\Sigma}})^{n/2}}$

where $\widehat{\boldsymbol{\Sigma}}$ is the maximum likelihood estimate of $\boldsymbol{\Sigma}$

The likelihood ratio for testing the submodel is thus

$\dfrac{(\det\widehat{\boldsymbol{\Sigma}}_H)^{-n/2}}{(\det\widehat{\boldsymbol{\Sigma}})^{-n/2}} = \left(\dfrac{\det\widehat{\boldsymbol{\Sigma}}}{\det\widehat{\boldsymbol{\Sigma}}_H}\right)^{n/2}$ which is

equivalent to Wilks's $\Lambda = \dfrac{\det\widehat{\boldsymbol{\Sigma}}}{\det\widehat{\boldsymbol{\Sigma}}_H} = \dfrac{\det\mathbf{W}}{\det\mathbf{T}}$

Note that in the univariate case, this is equal to $\dfrac{RSS}{RSS_H}$

which is equivalent to the usual F-ratio, as it is equivalent to

$$\frac{RSS_H}{RSS} = \frac{RSS + RSS_H - RSS}{RSS} = 1 + \frac{RSS_H - RSS}{RSS}$$

One can use then the (general) Bartlett approximation for likelihood ratios: $-m\log\Lambda$ is approximately $\chi^2_{p\nu_H}$

## Exact distribution for Wilks's $\Lambda$

If $X$ and $Y$ are independent and have both Gamma distribution with parameters $(\alpha_X, \gamma)$ and $(\alpha_Y, \gamma)$ respectively, then the ratio $X/(X+Y)$ has Beta distribution with parameters $(\alpha_X, \alpha_Y)$

A multivariate analog with Wishart distribution is: if $\mathbf{S}_1$ and $\mathbf{S}_2$ are independent and have Wishart distributions $\mathcal{W}_{n_1}(\boldsymbol{\Sigma})$ and $\mathcal{W}_{n_2}(\boldsymbol{\Sigma})$ respectively, then the distribution of

$$\Lambda = \frac{\det(\mathbf{S}_1)}{\det(\mathbf{S}_1 + \mathbf{S}_2)}$$

does not depend on $\boldsymbol{\Sigma}$, only on its dimension $p$ and $n_1, n_2$. It is known as *Wilks's* or also *multivariate Beta distribution*, as it is a product of $d$ independent random variables with Beta distribution

Its logarithm (or actually, $-\log$) can be handled numerically

...

## Example: dogs

5 groups:

```
> levels(as.factor(Canine$Group))
[1] "Cuon" "GoldenJackal" "IndianWolf"
[4] "ModernThaiDog" "PrehistoricThaiDog"
> Canine
   Sample  X1   X2 X3 X4 X5   X6 X7 X8  X9  Sex  Group
1       1 123 10.1 23 23 19  7.8 32 33 5.6  Male ModernThaiDog
2       2 137  9.6 19 22 19  7.8 32 40 5.8  Male ModernThaiDog
3       3 121 10.2 18 21 21  7.9 35 38 6.2  Male ModernThaiDog
4       4 130 10.7 24 22 20  7.9 32 37 5.9  Male ModernThaiDog
5       5 149 12.0 25 25 21  8.4 35 43 6.6  Male ModernThaiDog
...
> attach(Canine)
> nrow(Canine)
[1] 77
```

Is there a way to visualize this?

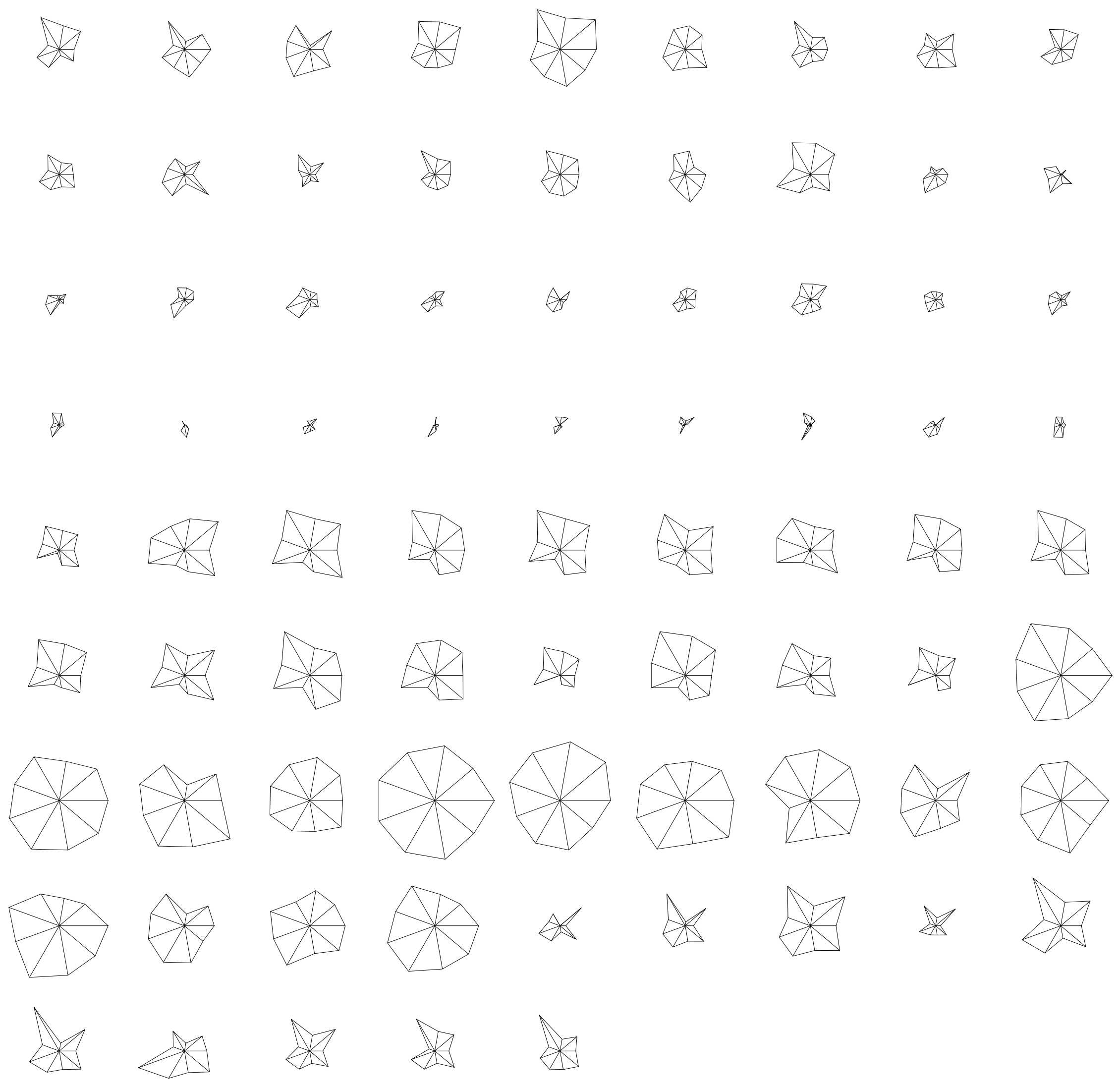
## Dogs: plotting (one way of)

```
> stars(Canine[,2:10])
> stars(Canine[,2:10],labels=Canine$Group,lwd=1)
> stars(Canine[,2:10],labels=as.numeric(as.factor(Canine$Group)),
+ lwd=1)
> stars(dogs[,2:10],lwd=0.7,
+ labels=strtrim(as.character(dogs$Group),6), cex=0.7, flip=F,
+ col.stars=
+ c("pink","grey","white")[as.numeric(as.factor(dogs$Sex))],ncol=7)
```

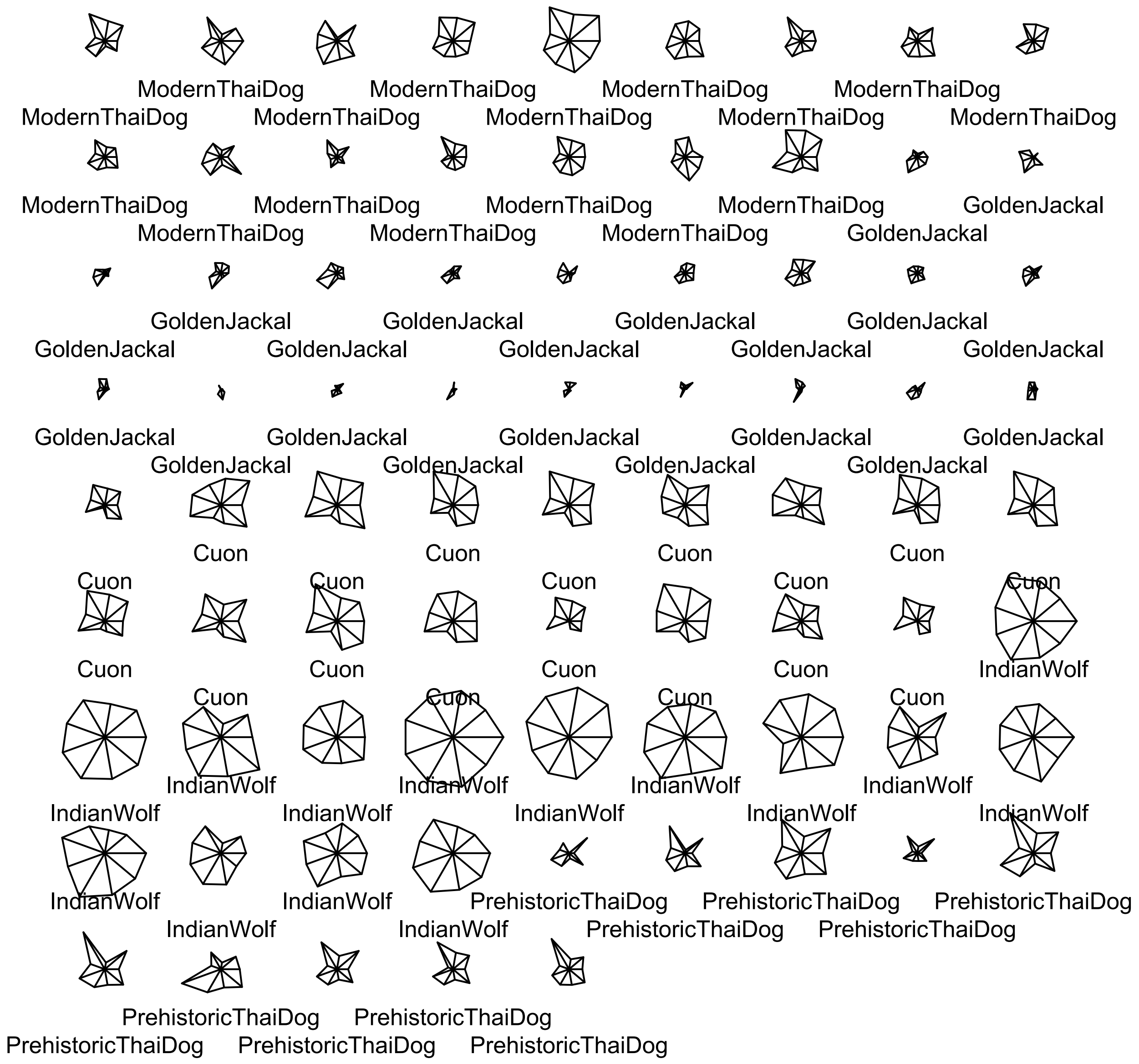
## Dogs: profiles (another plotting method)

```
> plot(c(1:9,1:9),
+ c(apply(Canine[,2:10],2,max),apply(Canine[,2:10],2,min)),
+ type='n',xlab='',ylab='')
> for (k in (1:77)[Group=="Cuon"])
+ lines(1:9,Canine[k,2:10],pch=1,type='b',col='black')
> for (k in (1:77)[Group=="GoldenJackal"])
+ lines(1:9,Canine[k,2:10],pch=2,type='b',col='yellow')
> for (k in (1:77)[Group=="IndianWolf"])
+ lines(1:9,Canine[k,2:10],pch=3,type='b',col='green')
> for (k in (1:77)[Group=="ModernThaiDog"])
+ lines(1:9,Canine[k,2:10],pch=4,type='b',col='magenta')
> for (k in (1:77)[Group=="PrehistoricThaiDog"])
+ lines(1:9,Canine[k,2:10],pch=5,type='b',col='blue')
```

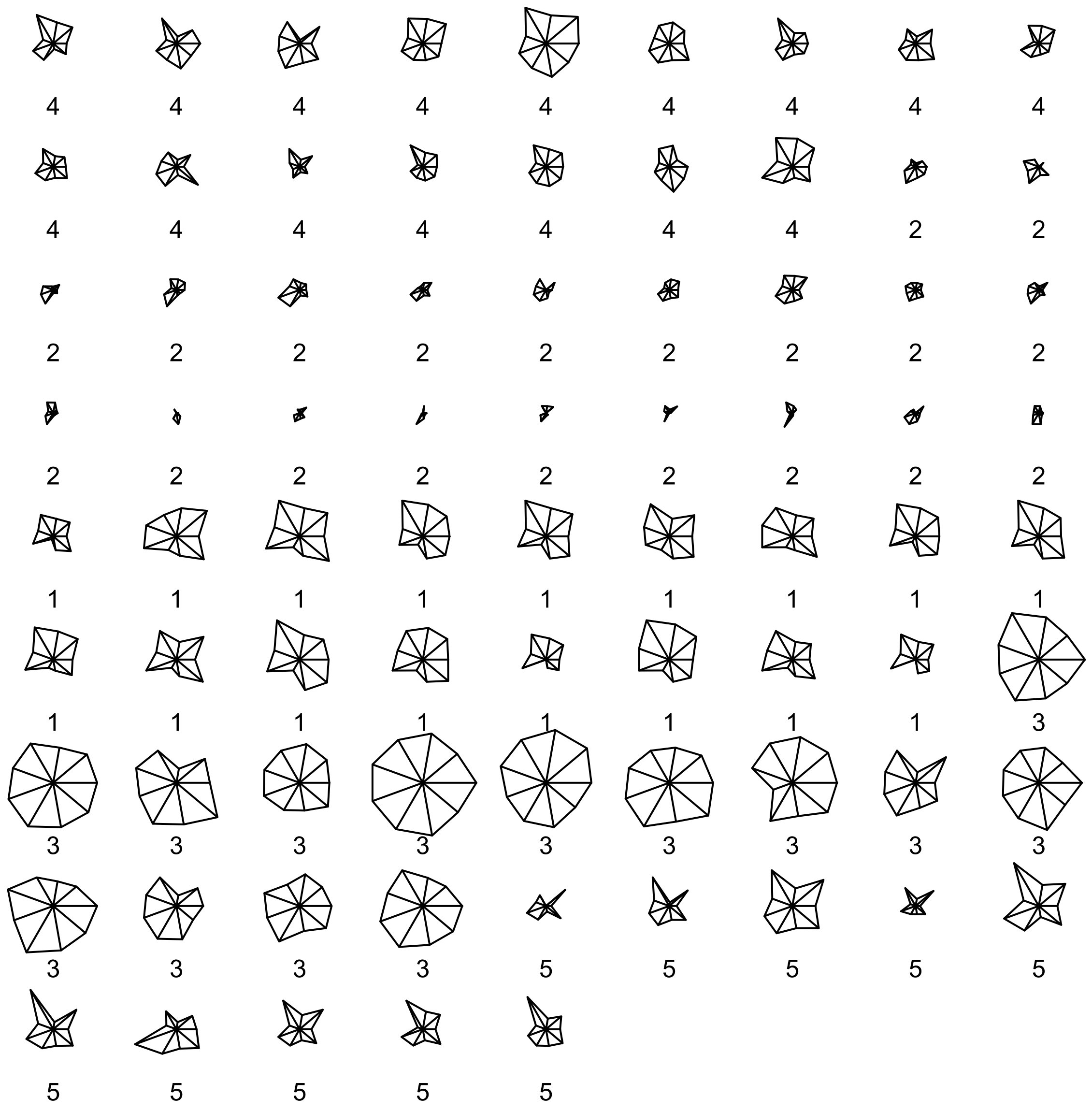
## Star plot: this is the default now



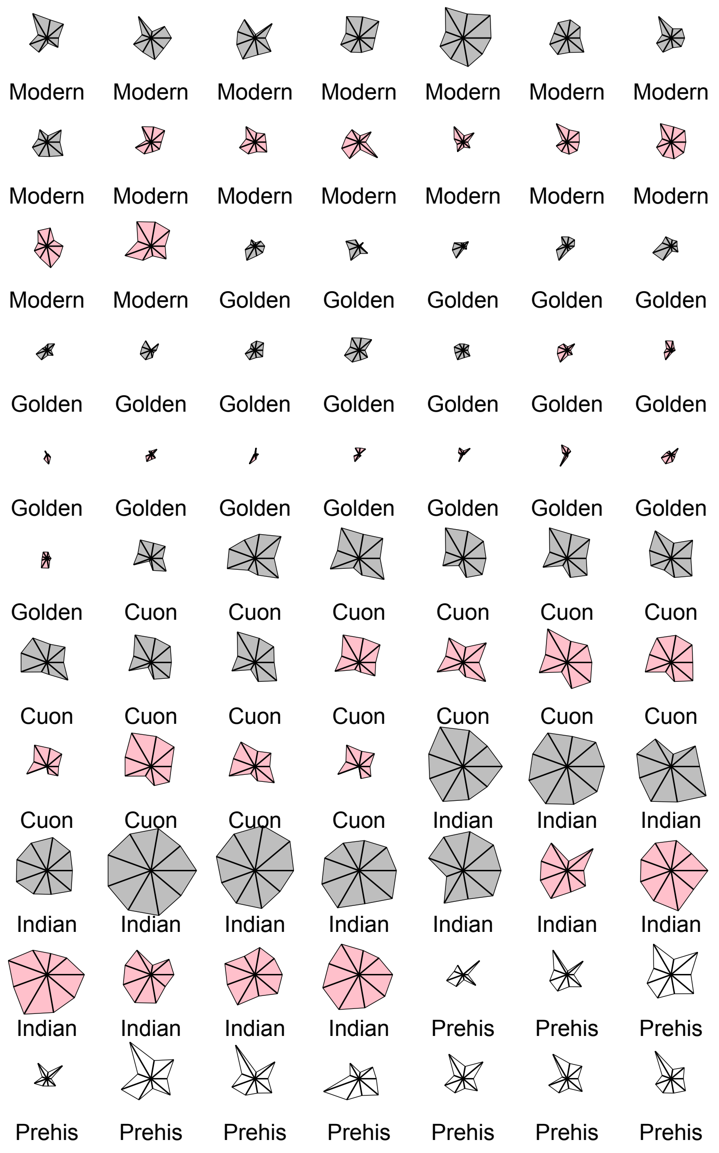
## Is this one better?



# Or this one?

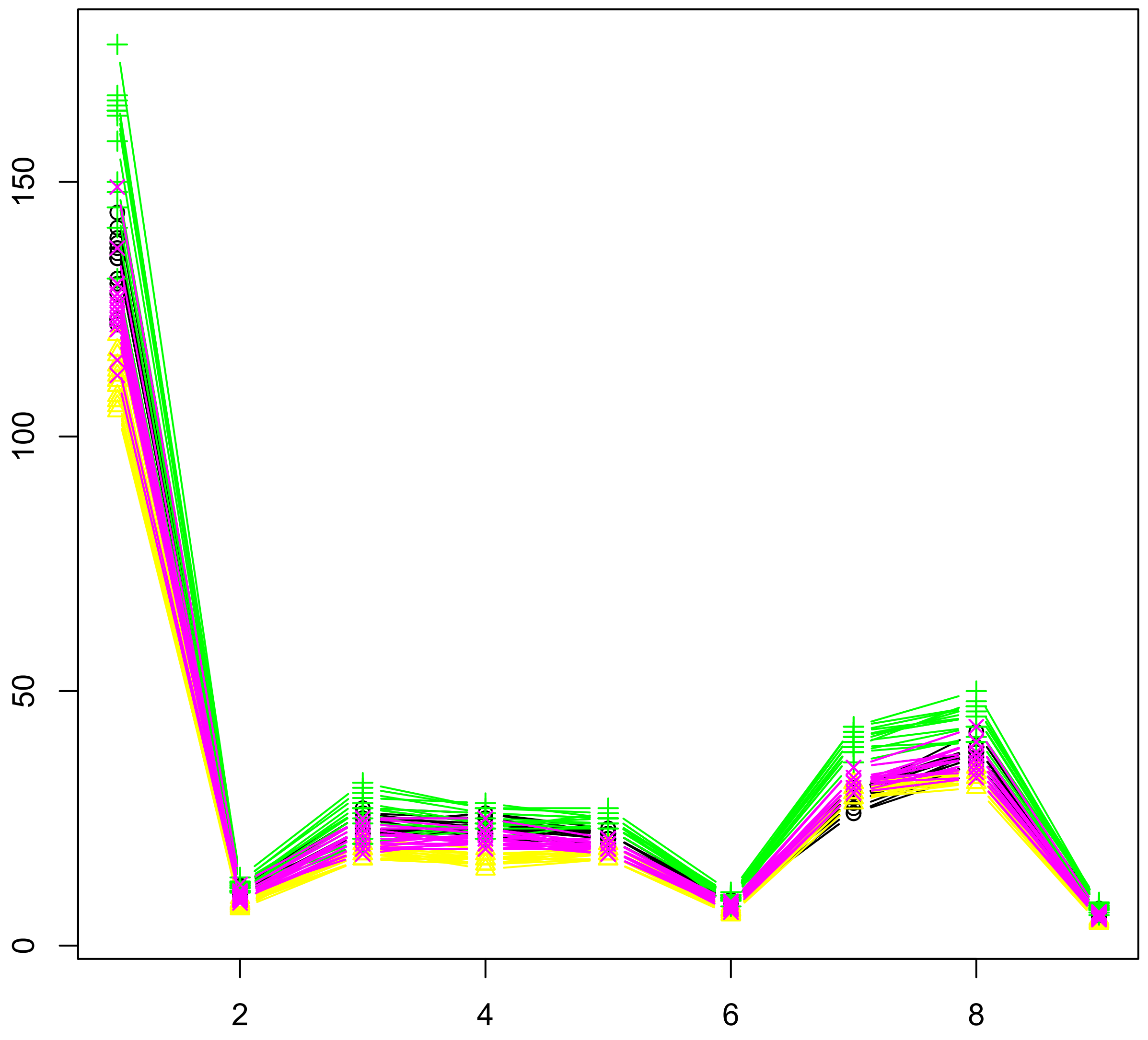

4 4 4 4 4 4 4 4 4
4 4 4 4 4 4 4 2 2
2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 3
3 3 3 3 3 3 3 3 3
3 3 3 3 5 5 5 5 5
5 5 5 5 5

## How about this one?

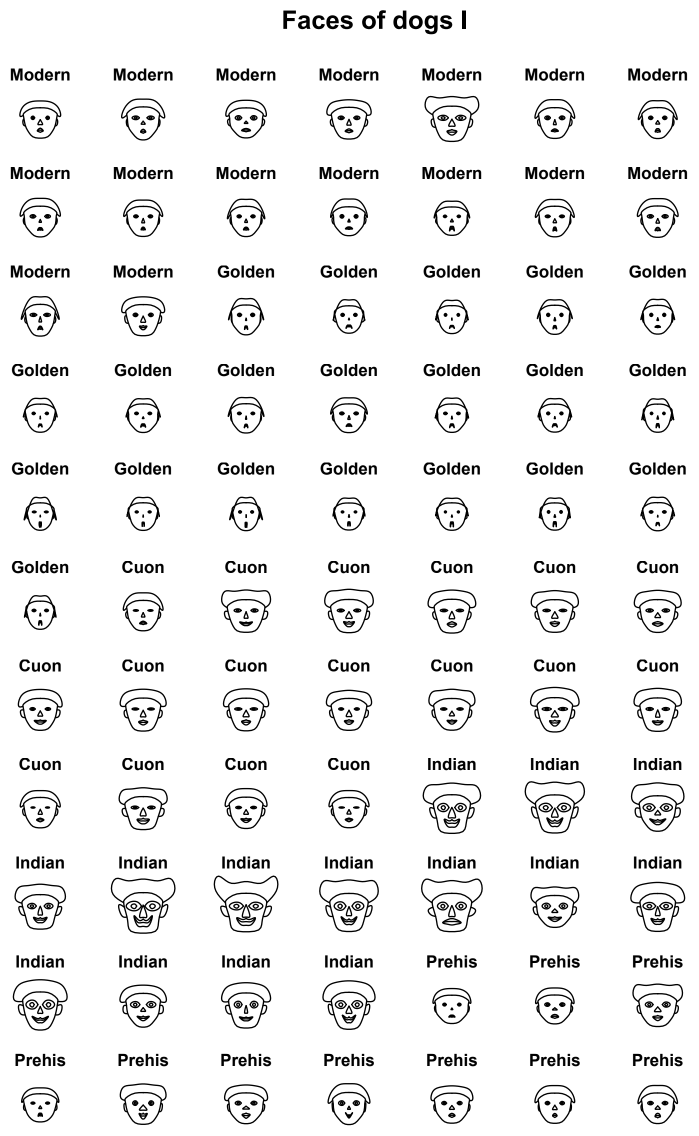


(Colors show sex: male, female, or unknown)

# Profiles

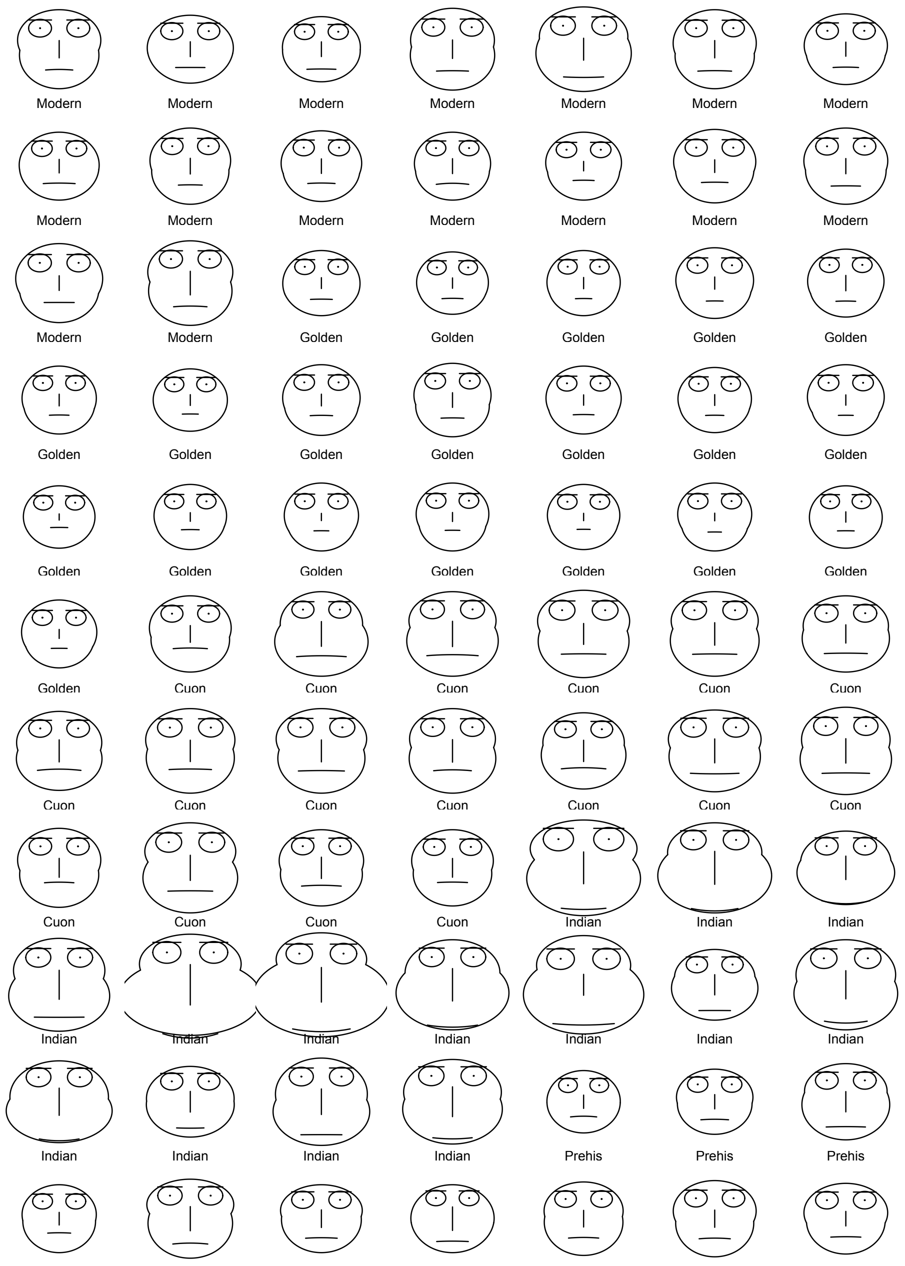


# Fancy plotting through glyphs: Chernoff faces



Visualizing variables by various facial characteristics

## More original look



For how to do it in R, google up "Chernoff faces R"

## And now the rigorous stuff

```
> respo <- cbind(X1,X2,X3,X4,X5,X6,X7,X8,X9)
> summary(manova(respo~Group, data=Canine), test='H')
          Df Hotelling-Lawley approx F num Df den Df    Pr(>F)
Group      4           25.129   43.627     36    250 < 2.2e-16 ***
Residuals  72

> summary(manova(respo~Group, data=Canine), test='R')
         Df     Roy approx F num Df den Df    Pr(>F)
Group     4  16.348  121.699      9     67 < 2.2e-16 ***
Residuals 72

> summary(manova(respo~Group, data=Canine), test='P')
          Df  Pillai approx F num Df den Df    Pr(>F)
Group      4  2.5892  13.6622     36    268 < 2.2e-16 ***
Residuals  72

> summary(manova(respo~Group, data=Canine), test='W')
             Df   Wilks approx F num Df den Df    Pr(>F)
Group      4.00  0.0022  27.6656  36.00 241.57 < 2.2e-16 ***
Residuals  72.00
```

## More sophisticated modeling possible

```
> summary(manova(respo~Group*Sex, data=Canine),test='H')
           Df Hotelling-Lawley approx F num Df den Df    Pr(>F)
Group       4           31.664   51.454     36    234 < 2.2e-16 ***
Sex         1            0.486    3.242      9     60  0.002866 **
Group:Sex   3            0.656    1.426     27    176  0.091126 .
Residuals  68
> summary(manova(respo~Group*Sex, data=Canine),test='R')
          Df     Roy approx F num Df den Df    Pr(>F)
Group      4  21.463  150.242      9     63 < 2.2e-16 ***
Sex        1   0.486    3.242      9     60  0.002866 **
Group:Sex  3   0.412    2.839      9     62  0.007281 **
Residuals 68
> summary(manova(respo~Group*Sex, data=Canine),test='P')
           Df  Pillai approx F num Df den Df    Pr(>F)
Group       4  2.6446  13.6581     36    252 < 2.2e-16 ***
Sex         1  0.3272   3.2417      9     60  0.002866 **
Group:Sex   3  0.5054   1.3958     27    186  0.103601
Residuals  68
> summary(manova(respo~Group*Sex, data=Canine),test='W')
              Df   Wilks approx F num Df den Df    Pr(>F)
Group       4.00  0.0014  30.2015  36.00 226.59 < 2.2e-16 ***
Sex         1.00  0.6728   3.2417   9.00  60.00  0.002866 **
Group:Sex   3.00  0.5637   1.4125  27.00 175.87  0.096985 .
Residuals  68.00
```

## And for two-level one-way layout all coincide

... and reduce to two-sample Hotelling $T^2$.

```
> summary(manova(respo~Sex,
+ data=Canine,subset=Sex!="Unknown"),test='H')
          Df Hotelling-Lawley approx F num Df den Df Pr(>F)
Sex        1          0.14886  0.94279      9     57 0.4961
Residuals 65
> summary(manova(respo~Sex,
+ data=Canine,subset=Sex!="Unknown"),test='R')
          Df     Roy approx F num Df den Df Pr(>F)
Sex        1 0.14886  0.94279      9     57 0.4961
Residuals 65
> summary(manova(respo~Sex,
+ data=Canine,subset=Sex!="Unknown"),test='P')
          Df  Pillai approx F num Df den Df Pr(>F)
Sex        1 0.12957  0.94279      9     57 0.4961
Residuals 65
> summary(manova(respo~Sex,
+ data=Canine,subset=Sex!="Unknown"),test='W')
          Df   Wilks approx F num Df den Df Pr(>F)
Sex        1 0.87043  0.94279      9     57 0.4961
Residuals 65
```

## Still Extending Classics: Two-Sample Hotelling's $T^2$ (Now As a Special Case of MANOVA)

## And now: two-sample, first again univariate

Two-sample t-test: if

$y_{1i}$, $i = 1, \ldots, n_1$ are modeled iid with $N(\mu_1, \sigma_1^2)$, and

$y_{2i}$, $i = 1, \ldots, n_2$ are modeled iid with $N(\mu_2, \sigma_2^2)$, and

it is believed that those random variables are all independent

(between, but also within groups - “samples” - as well)

and it is also believed that $\sigma_1^2 = \sigma_2^2$

then $\dfrac{\bar{x}_1 - \bar{x}_2}{s_{\text{pooled}}} \sqrt{\dfrac{n_1 n_2}{n_1 + n_2 - 2}}$ is modeled by $t_{n_1+n_2-2}$,

or equivalently, $\left(\dfrac{n_1 n_2}{n_1 + n_2 - 2}\right) \dfrac{(\bar{x}_1 - \bar{x}_2)^2}{s^2_{\text{pooled}}}$ by $F_{1,n_1+n_2-2}$.

The important component here is the “pooled” estimate of the common variance

$$s^2_{\text{pooled}} = \frac{1}{n_1 + n_2 - 2} \left( \sum_{i=1}^{n_1} (x_{1i} - \bar{x}_1)^2 + \sum_{i=1}^{n_2} (x_{2i} - \bar{x}_2)^2 \right)$$

## Two-sample: multivariate

Hotelling $T^2$: if

$\mathbf{y}_{1i}$, $i = 1, \ldots, n_1$ are modeled iid with $N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$, and

$\mathbf{y}_{2i}$, $i = 1, \ldots, n_2$ are modeled iid with $N(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$, and

those random vectors are believed to be are independent

(between and within groups)

(but not necessarily within vectors)

and $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2$, then (writing $\boldsymbol{\mu} = \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2$)

$$\frac{n_1 + n_2 - 1 - p}{(n_1 + n_2 - 2)p} \; \frac{n_1 n_2}{n_1 + n_2}(\bar{\mathbf{y}}_1 - \bar{\mathbf{y}}_2 - \boldsymbol{\mu})^T \mathbf{S}_{\text{pooled}}^{-1}(\bar{\mathbf{y}}_1 - \bar{\mathbf{y}}_2 - \boldsymbol{\mu})$$

can be considered $F_{p, n_1+n_2-1-p}$ or also

$$T^2 = \frac{n_1 n_2}{n_1 + n_2}(\bar{\mathbf{y}}_1 - \bar{\mathbf{y}}_2 - \boldsymbol{\mu})^T \mathbf{S}_{\text{pooled}}^{-1}(\bar{\mathbf{y}}_1 - \bar{\mathbf{y}}_2 - \boldsymbol{\mu})$$

can be considered $\frac{(n_1 + n_2 - 2)p}{n_1 + n_2 - 1 - p} F_{p, n_1+n_2-1-p}$

## The pooled estimate of $\boldsymbol{\Sigma}$

The estimate of the common variance-covariance matrix is analogously the $p \times p$ matrix $\mathbf{S}_{\text{pooled}}$ whose element in $k$-th row and $\ell$-th column is

$$\frac{1}{n_1 + n_2 - 2}\left(\sum_{i=1}^{n_1}(y_{1ik} - \bar{y}_{1k})(y_{1i\ell} - \bar{y}_{1\ell}) + \sum_{i=1}^{n_2}(y_{2ik} - \bar{y}_{2k})(y_{2i\ell} - \bar{y}_{2\ell})\right)$$

## Applications akin to univariate ones

**Two-sample comparisons**

testing $\mathrm{H}_0 \colon \mu_1 = \mu_2$.

**Two-sample confidence intervals**

for $\boldsymbol{\mu} = \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2$

## Truly multivariate application: profile analysis

Profile analysis is again more general:

we test $\quad H_0 : \mathbf{C}\boldsymbol{\mu}_1 = \mathbf{C}\boldsymbol{\mu}_2$, where $\mathbf{C}$ is $k \times p$

using $\quad T^2 = \frac{n_1 n_2}{n_1 + n_2}(\bar{\mathbf{y}}_1 - \bar{\mathbf{y}}_2)^T \mathbf{C}^T [\mathbf{C}\mathbf{S}_{\text{pooled}}\mathbf{C}^T]^{-1}\mathbf{C}(\bar{\mathbf{y}}_1 - \bar{\mathbf{y}}_2)$

with the distribution $\quad \frac{(n_1 + n_2 - 2)k}{n_1 + n_2 - 1 - k} F_{k, n_1+n_2-1-k}$

## Example

V1, V2 and V3, V4 are the same measurements made in two labs.

```
> eff
   V1  V2 V3 V4
1   6  27 25 15
2   6  23 28 13
3  18  64 36 22
4   8  44 35 29
5  11  30 15 31
6  34  75 44 64
7  28  26 42 30
8  71 124 54 64
9  43  54 34 56
10 33  30 29 20
11 20  14 39 21
```

## Paired $T^2$

Let's try the paired Hotelling first:

```
> summary(manova(cbind(V1-V3,V2-V4)~1,data=eff),
+ intercept=T,test="H")
            Df Hotelling-Lawley approx F num Df den Df  Pr(>F)
(Intercept)  1           1.3639   6.1377      2      9 0.02083 *
Residuals   10
> 1-pf(13.6*9/20,2,9)
[1] 0.02098437
```

## And two-sample $T^2$

And then the two-sample Hotelling:

```
> respon=cbind(c(eff$V1,eff$V3),c(eff$V2,eff$V4))
> sam=factor(rep(c(0,1),c(11,11)))
> sam
 [1] 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1
Levels: 0 1
> summary(manova(respon~samp),test="H")
          Df Hotelling-Lawley approx F num Df den Df   Pr(>F)
sam        1           0.6332   6.0158      2     19 0.009463 **
Residuals 20
```

## Review of t-tests: one-sample

Notation: $\bar{y} = \frac{1}{n}\sum_i y_i, \qquad s^2 = \frac{1}{n-1}\sum_i (y_i - \bar{y})^2$

the *t statistic* $\sqrt{n}\,\frac{\bar{y}-\mu}{s}$ is $t_{n-1}$ (abusing the notation)

if $y_1, y_2, \ldots, y_n$ are iid, each is $N(\mu, \sigma^2)$; then it follows that

- $\bar{y}$ is $N(\mu, \frac{1}{n}\sigma^2)$, and then $\frac{\bar{y}-\mu}{\frac{\sigma}{\sqrt{n}}} = \sqrt{n}\,\frac{\bar{y}-\mu}{\sigma}$ is $N(0,1)$
- $\frac{(n-1)s^2}{\sigma^2} = \frac{\sum_i (x_i - \bar{x})^2}{\sigma^2}$ is $\chi^2_{n-1}$
- $\bar{x}$ and $s^2$, and thus $\sqrt{n}\,\frac{\bar{x}-\mu}{\sigma}$ and $\frac{(n-1)s^2}{\sigma^2}$ are independent

which together yields the $t_{n-1}$ distribution of

$$\frac{\sqrt{n}\,\frac{\bar{x}-\mu}{\sigma}}{\sqrt{\frac{\frac{(n-1)s^2}{\sigma^2}}{n-1}}} = \sqrt{n}\,\frac{\bar{x}-\mu}{s}$$

For large $n$, this distribution is approximately $N(0,1)$

# Two sample (t) test: univariate

If $y_{1i}$, $i = 1, \dots, n_1$ are iid with $N(\mu_1, \sigma_1^2)$,

and $y_{2i}$, $i = 1, \dots, n_2$ are iid with $N(\mu_2, \sigma_2^2)$,

and they are independent also between samples, and $\sigma_1^2 = \sigma_2^2$,

then $\dfrac{\bar{y}_1 - \bar{y}_2}{\bar{s}}\sqrt{\dfrac{n_1 n_2}{n_1 + n_2 - 2}}$ is $t_{n_1+n_2-2}$ ($\bar{s}$ is "pooled" estimate)

or, alternatively, $\left(\dfrac{n_1 n_2}{n_1 + n_2 - 2}\right)\dfrac{(\bar{y}_1 - \bar{y}_2)^2}{\bar{s}^2}$ is $F_{1,n_1+n_2-2}$.

## Two sample ($T^2$) test: multivariate

If $\mathbf{y}_{1i}$, $i = 1, \dots, n_1$ are iid with $N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$,

and $\mathbf{y}_{2i}$, $i = 1, \dots, n_2$ are iid with $N(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$,

and they are independent also between samples,

and $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2$,

then $(\boldsymbol{\mu} = \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$

$$\frac{n_1 + n_2 - 1 - p}{(n_1 + n_2 - 2)p} \; \frac{n_1 n_2}{n_1 + n_2}(\bar{\mathbf{y}}_1 - \bar{\mathbf{y}}_2 - \boldsymbol{\mu})^T \mathbf{S}_{\text{pooled}}^{-1} (\bar{\mathbf{y}}_1 - \bar{\mathbf{y}}_2 - \boldsymbol{\mu})$$

is $F_{p, n_1 + n_2 - 1 - p}$, or, alternatively,

$$T^2 = \frac{n_1 n_2}{n_1 + n_2}(\bar{\mathbf{y}}_1 - \bar{\mathbf{y}}_2 - \boldsymbol{\mu})^T \mathbf{S}_{\text{pooled}}^{-1} (\bar{\mathbf{y}}_1 - \bar{\mathbf{y}}_2 - \boldsymbol{\mu})$$

is $\dfrac{(n_1 + n_2 - 2)p}{n_1 + n_2 - 1 - p} F_{p, n_1 + n_2 - 1 - p}$

where $\mathbf{S}_{\text{pooled}}$ is the pooled sample variance-covariance matrix with elements

$$\frac{1}{n_1 + n_2 - 2}\left( \sum_{i=1}^{n_1} (y_{1ik} - \bar{y}_{1k})(y_{1i\ell} - \bar{y}_{1\ell}) + \sum_{i=1}^{n_2} (y_{2ik} - \bar{y}_{2k})(y_{2i\ell} - \bar{y}_{2\ell}) \right)$$

## Dual View - Reconciling Distances: Multidimensional Scaling (Ordination)

## The principle

Imagine that you have a map, with the distances of cities
and then you lose it, and what remains are distances
and you have to reconstruct the map

# Fairmont hotels in Alberta

Caution: only to illustrate methodology - otherwise not quite an example of a real data analysis. Driving distances:

```
> Fairorg    ## original data out of travel claims
           Banff Calgary Edmonton Jasper LakeLouise
Banff          0     128       NA     NA         58
Calgary      128       0      294     NA         NA
Edmonton      NA     294        0    370         NA
Jasper        NA      NA      370      0        229
LakeLouise    58      NA       NA    229          0
> Fairmont   ## completed by "minimal sum"
           Banff Calgary Edmonton Jasper LakeLouise
Banff          0     128      422    287         58
Calgary      128       0      294    415        186
Edmonton     422     294        0    370        480
Jasper       287     415      370      0        229
LakeLouise    58     186      480    229          0

> plot(cmdscale(Fairmont),
+ pch=16,xlab='',ylab='',xlimrrank=c(-250,350),ylim=c(-250,200))
> text(cmdscale(Fairmont)-15,lab=row.names(Fairmont))
```

## Banff

## Edmonton

## Calgary

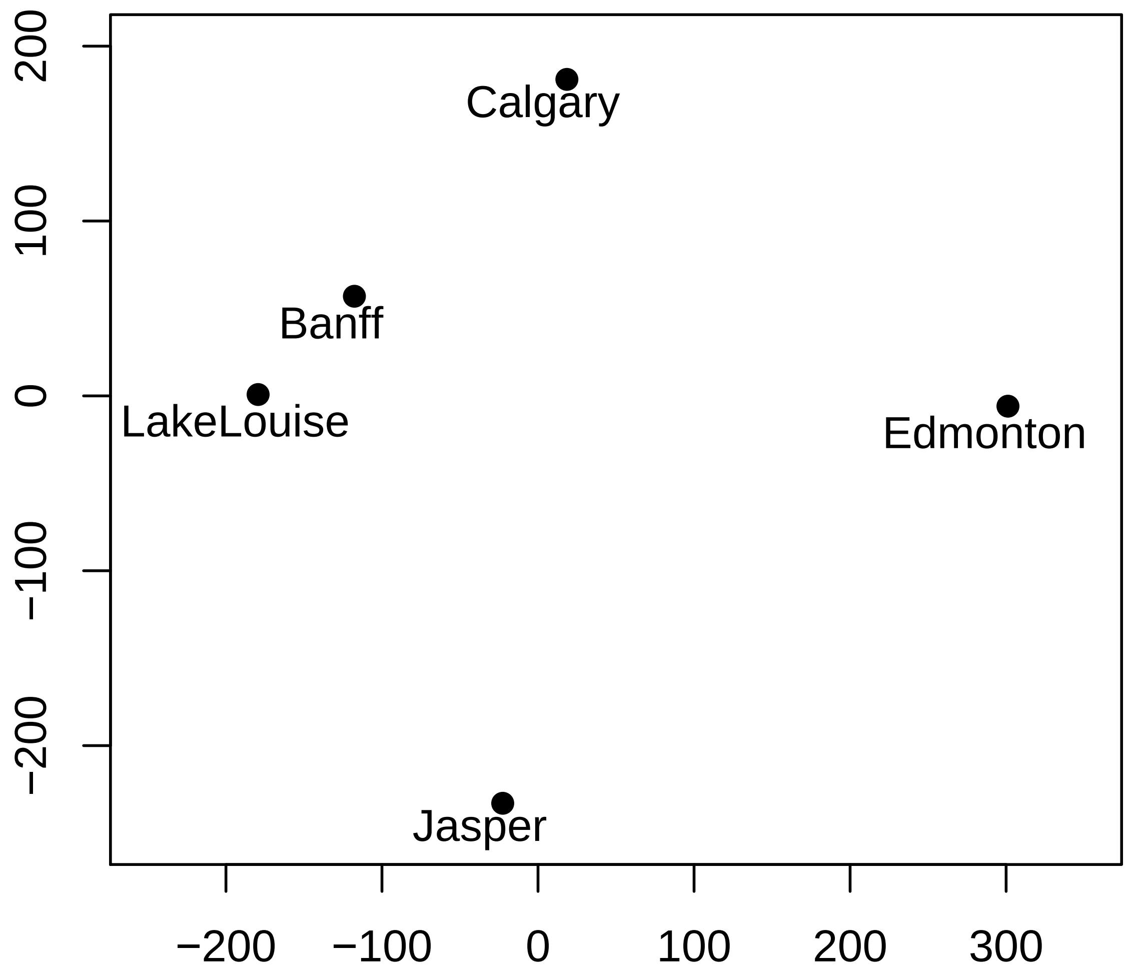
## Lake Louise

## Jasper

## And now something different: Skoki lodge

## Even the king goes there by foot

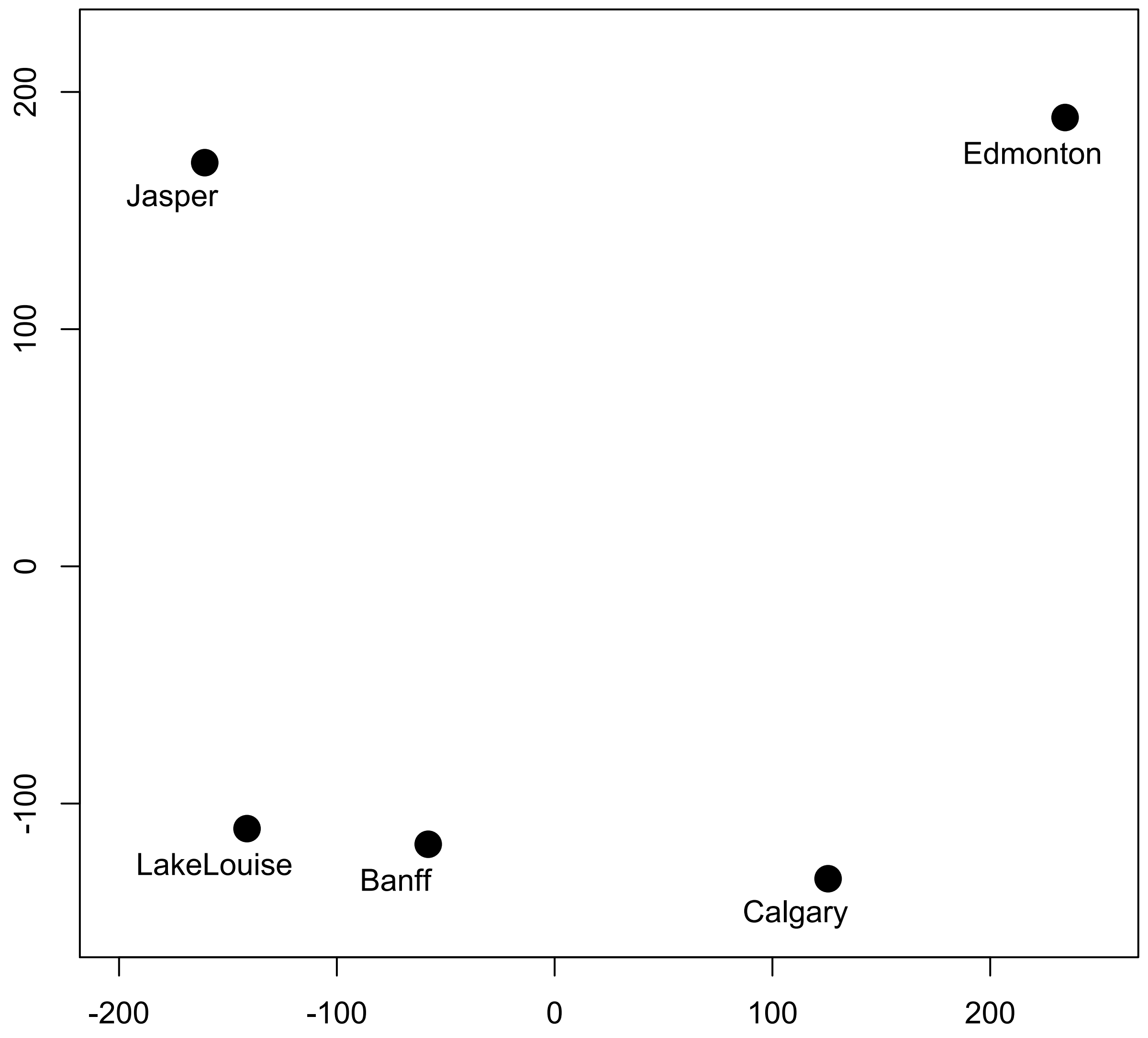
# Is this Alberta?



## This is Alberta

ALBERTA
Edmonton
Calgary
Red Deer
Jasper National Park of Canada
Banff National Park
Wood Buffalo National Park of Canada
Fort McMurray
Grande Prairie
Lethbridge
Medicine Hat

## And now?



```
rbind(c(cos(pi*0.21),sin(pi*0.21)),c(sin(pi*0.21),-cos(pi*0.21)))
```

# OK, let's try something closer...

| Dojezdová vzdálenost (kilometr) | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Brno | 1 | | 296 | 77 | 307 | 216 | 168 | 296 | 146 | 96 | 169 | 337 | 93 | 208 | 176 | 240 |
| Plzeň | 2 | 296 | | 369 | 200 | 137 | 208 | 181 | 213 | 388 | 462 | 81 | 217 | 92 | 469 | 100 |
| Olomouc | 3 | 77 | 370 | | 236 | 290 | 139 | 370 | 135 | 63 | 98 | 411 | 167 | 282 | 104 | 314 |
| Liberec | 4 | 309 | 198 | 237 | | 249 | 98 | 96 | 155 | 401 | 434 | 233 | 230 | 110 | 449 | 135 |
| České Budějovice | 5 | 217 | 137 | 290 | 250 | | 253 | 239 | 258 | 309 | 383 | 218 | 138 | 151 | 389 | 183 |
| Hradec Králové | 6 | 167 | 209 | 140 | 97 | 253 | | 190 | 24 | 206 | 238 | 240 | 112 | 116 | 245 | 143 |
| Ústí nad Labem | 7 | 298 | 181 | 371 | 96 | 238 | 190 | | 196 | 390 | 463 | 122 | 219 | 93 | 470 | 87 |
| Pardubice | 8 | 146 | 214 | 135 | 154 | 193 | 24 | 196 | | 201 | 234 | 246 | 90 | 121 | 240 | 149 |
| Zlín | 9 | 96 | 389 | 62 | 400 | 308 | 206 | 389 | 202 | | 124 | 430 | 186 | 300 | 131 | 333 |
| Ostrava | 10 | 170 | 463 | 94 | 440 | 383 | 239 | 463 | 235 | 125 | | 504 | 261 | 375 | 14 | 407 |
| Karlovy Vary | 11 | 337 | 82 | 410 | 234 | 218 | 241 | 121 | 246 | 429 | 502 | | 258 | 127 | 509 | 105 |
| Jihlava | 12 | 93 | 218 | 167 | 229 | 138 | 113 | 218 | 90 | 186 | 259 | 259 | | 130 | 266 | 162 |
| Praha | 13 | 209 | 91 | 282 | 109 | 150 | 116 | 92 | 121 | 301 | 375 | 128 | 130 | | 382 | 31 |
| Paskov | 14 | 176 | 469 | 100 | 451 | 388 | 244 | 469 | 240 | 131 | 13 | 510 | 266 | 380 | | 412 |
| Kladno | 15 | 240 | 99 | 313 | 135 | 181 | 143 | 86 | 148 | 332 | 406 | 105 | 161 | 31 | 412 | |

## But rather better example first: Crimes

After interviewing respondents, we have the following table of **dissimilarities** between various crimes

```
 0
 2  0
15 13  0
15 14  6  0
15 14  3  3  0
 6  4  3 10 12  0
 4  2 14 14 15 13  0
15  6  5 11 10  6 10  0
15 10 12  2  2 12 14  7  0
 2  2 15 15 15 14  4 11 15  0
14 12  2 11 11  6 15 11 12 14  0
14 15  8  4  3 12 14 11  3 15 11  0
 9 13  6 14  9  5 10 11  7 13  7 11 0

"Murder" "Manslaugh" "Burglary" "Drugs" "Shoplift" "Drunkdrive" "Arson"
"Bodyharm" "Drunkdisord" "Rape" "Cartheft" "Trespass" "Taxes"
```

## The story of multidimensional scaling

Can we summarize these data somehow?

Say, plot them as points on the plane?

**Multidimensional scaling**: we would like to represent data in a lower (often 2-) dimensional (hyper)plane while preserving the dissimilarities (distances) between the data points as much as possible (minimizing the possible distortion caused by the reduction of dimensionality).

Alternative name: ordination

## Two principal attitudes

Classical - also called **metric** multidimensional scaling:

- works rather with Euclidean distances than anything else
- considers pictorial projections into the lower dimensional hyperplanes rather than general (possibly nonlinear) mappings

Nonlinear - also called **non-metric** multidimensional scaling

- works with general, even often non-metric distances
- considers all, posiibly also non-linear mappings into the lower dimensional hyperplanes; finds them via numerical algorithm optimizing certain criterion of optimality

## General dissimilarities $\approx$ (more or less) distances

The prefix “dis-” means that similar objects would have low dissimilarity and objects not similar high - if it were other way round the name would rather be “similarity”

What constitutes a “distance”? Some desirable properties:

D1. *nonnegativity*: from A to B it is $\geqslant 0$

D2. *symmetry*: from A to B = from B to A

D3. *identification*: from A to A it should be 0

These are usually the minimal properties which we require for “distance” or **dissimilarity**. A **similarity** has scale reversed: it satisfies 1. and 2. but instead of 3. it is required that

S. similarity of A and B is always $\leqslant$ than similarity of A and A - and thus also $\leqslant$ than similarity of B and B

## More stringent properties

Possible more stringent properties of dissimilarities are

D4. *triangle inequality*

$$\text{from A to C} \leqslant \text{from A to B} + \text{from B to C}$$

A dissimilarity satisfying D1-D4 is called a **pseudometric**. A pseudometric is a **metric**, if it also satisfies

D5. if from A to B is 0, then A $=$ B

If a dissimilarity satisfies D5 and

D6. from A to C $\leqslant \max\{\text{from A to B}, \text{from B to C}\}$

(note that the latter implies triangle inequality, hence D5 and D6 imply a metric), it is called an **ultrametric**. We will see an example in hierarchical clustering, where it arises as a distance on a tree.

## Aspects of dissimilarities

Sometimes we may like dissimilarities ranging from 0 to 1, sometimes from 0 to $+\infty$; sometimes we need to convert a similarity measure to a dissimilarity; useful transformations are $1 - d$, $1/d$, $1/(1 + d)$

Another way to transform a similarity $c_{ij}$ into a dissimilarity is to consider the dissimilarity

$$d_{ij} = (c_{ii} - 2c_{ij} + c_{jj})^{1/2}$$

If we are calculating distances from the datapoints, we may want them standardized: all variables mean zero, standard deviation one.

Dissimilarities may be computed from variables, or given explicitly; they can be conveniently organized into a matrix - a symmetric one if the dissimilarities are so.

In R, it is a function: `dist()`

Also, in the widely used R package `cluster` it is `daisy()`

## Examples: metrics

Euclidean metric:

$$d_2(A, B) = \sqrt{(A_1 - B_1)^2 + \cdots + (A_p - B_p)^2}$$

Manhattan (“Edmonton”, cityblock) metric

$$d_1(A, B) = |A_1 - B_1| + \cdots + |A_p - B_p|$$

Maximum metric

$$d_\infty(A, B) = \max\{|A_1 - B_1|, \ldots, |A_p - B_p|\}$$

These are all special cases (limit for $p = \infty$) of the Minkowski distance

$$d_p(A, B) = (|A_1 - B_1|^p + \cdots + |A_p - B_p|^p)^{1/p}$$

## Some dissimilarities aimed at other data types

For binary data, dissimilarities may be based on various association coefficients known from contingency tables, like Yule's $(ad - bc)/(ad + bc)$.

If we interpret contingency table as presence-absence matching, then we have things like $a + d/(a + b + c + d)$ (ordinary matching), $a/\sqrt{(a + b)(a + c)}$ (Ochiai index), or $2a/(2a + b + c)$ (Dice-Sorensen index), or $a/(a + b + c)$ (Jaccard index).

For ordinal data (rankings), dissimilarities may be based on ranks

Canberra metric $$\frac{n}{n^*}\sum_i \frac{|x_i - y_i|}{|x_i| + |y_i|}$$

is aimed at counts, that is, nonnegative $x_i$ and $y_i$

hence sometimes also $|x_i + y_i|$ or $x_i + y_i$ in the denominator

the sum is over the $n^*$ terms that are *not* of the form 0/0

(if all have this form, then undefined)

## Examples of dissimilarities aimed at 0-1 data

Hamming metric - handy for nominal (categorical) data:

$1-$ proportion of $i$ such that $x_i = y_i$

proportion means: the number of the $i$'s for which it is true

divided by the total number of the $i$'s

equivalent: proportion of $i$ such that $x_i \neq y_i$

R function `dist()` has a distance "binary" defined also as

proportion of $i$ such that $x_i \neq y_i$

but the proportion is only among those that are not both 0

## Penrose's measure

There are measures of distances between groups (populations and samples), when the information is available only on means, standard deviations, and covariances.

Penrose's measure: suppose there are $p$ variables and several groups: the mean of $k$-th variable in $i$-th group is $\mu_{ki}$. If we know that all groups have variance $\nu_k$ of $k$-th variable, then

$$P_{ij} = \sum_{k=1}^{p} \frac{(\mu_{ki} - \mu_{kj})^2}{p\nu_k}$$

Does not include covariances: may overemphasize correlated variables.

## Mahalanobis distance

Euclidean distance after normalizing the data

If we know (or estimate) the variance-covariance matrix $\boldsymbol{\Sigma}$, we form its inverse $\boldsymbol{\Sigma}^{-1}$, whose elements are $\sigma^{rs}$. Then

$$D_{ij} = \sum_{r=1}^{p} \sum_{s=1}^{p} (\mu_{ri} - \mu_{rj}) \sigma^{rs} (\mu_{si} - \mu_{sj}) = (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^{\top} \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)$$

where $\boldsymbol{\mu}_i$ is the column vector with components $\mu_{ki}$.

## Distances based on proportions

If $p_k$, $q_k$ are proportions of the $k$-th attribute in the first and second group respectively, then we may take

$$d = \textstyle\sum_k \frac{1}{2}|p_k - q_k| \qquad \text{or } d = 1 - \frac{\displaystyle\sum_k p_k q_k}{\left(\displaystyle\sum_k p_k^2 \sum_k q_k^2\right)^{1/2}}$$

(we typically want them to range from 0 to 1)

## Classical (metric) multidimensional scaling: Euclidean distances

**Classical** (also called **metric**) **multidimensional scaling** is derived under the assumption that the distances are *Euclidean* - that is, they arose as the Euclidean distances among points in some high-dimensional space.

The crucial property that distinguishes the Euclidean metric from others is its relationship to the inner product. If the dissimilarities $d_{ij}$ between $n$ objects are Euclidean, then there is a collection of points, vectors $\mathbf{y}_i \in \mathbb{R}^p$ for some $p$ (not greater than $n-1$) such that

$$d_{ij}^2 = (\mathbf{y}_i - \mathbf{y}_j)^\mathsf{T}(\mathbf{y}_i - \mathbf{y}_j) = \mathbf{y}_i^\mathsf{T}\mathbf{y}_i + \mathbf{y}_j^\mathsf{T}\mathbf{y}_j - 2\mathbf{y}_i^\mathsf{T}\mathbf{y}_j = \|\mathbf{y}_i\|^2 + \|\mathbf{y}_j\|^2 - 2\mathbf{y}_i^\mathsf{T}\mathbf{y}_j$$

We do not need $\mathbf{y}_{ij}$ themselves - but we need to know the inner products $\mathbf{y}_i^\mathsf{T}\mathbf{y}_j$, and the question is how to recover those from given distances $d_{ij}$

## Recovering the inner product

Given the $d_{ij}$'s, let **A** be the matrix with the elements $-\frac{1}{2}d_{ij}^2$

If the dissimilarities are Euclidean, this matrix must be symmetric, have diagonal elements zero, and off-diagonal elements nonzero. As the Euclidean distances are invariant with respect to translations and rotations, we can choose without loss of generality the coordinate system so that $\mathbf{0} = \bar{\mathbf{y}}$, that is, $\mathbf{0} = \sum_{i=1}^{n} \mathbf{y}_i$

Now, given the matrix **A**, we construct the matrix **B** consisting of elements

$$b_{ij} = -\frac{1}{2}d_{ij}^2 + \frac{1}{2n}\sum_i d_{ij}^2 + \frac{1}{2n}\sum_j d_{ij}^2 - \frac{1}{2n^2}\sum_{i,j} d_{ij}^2$$

(for each element $a_{ij}$ of **A**, we subtract the average of its row, the average of its column, and add the average of the whole **A**)

This matrix is always symmetric if the dissimilarities are symmetric; if they are also Euclidean, the elements of **B** are equal to $\mathbf{y}_i^\top \mathbf{y}_j$

## Inner product recovered

Thus, $\mathbf{B}$ "reconstructs" the inner products of the $\mathbf{y}_i$'s - without necessarily knowing them, only their distances

In particular, if $\mathbf{y}_i^\top$ are lines of a matrix $\mathbf{Y}$, then $\mathbf{B} = \mathbf{Y}\mathbf{Y}^\top$

Conversely, if $\mathbf{B}$ is a symmetric and nonnegative definite $n \times n$ matrix, then it defines an inner product among some collection of $\mathbf{y}_i$, $i = 1, 2, \ldots, n$

- which in turn yields squared distances among the objects corresponding to some $\mathbf{y}_i$, as given above

## The proof

For all $i$ and $j$ we have

$$d_{ij}^2 = d_{ji}^2 = (\mathbf{y}_i - \mathbf{y}_j)^\top(\mathbf{y}_i - \mathbf{y}_j) = \|\mathbf{y}_i\|^2 + \|\mathbf{y}_j\|^2 - 2\mathbf{y}_i^\top \mathbf{y}_j$$

Using the fact that $\sum_i \mathbf{y}_i = \mathbf{0}$, we obtain

$$\frac{1}{n}\sum_i d_{ij}^2 = \frac{1}{n}\sum_i \|\mathbf{y}_i\|^2 + \|\mathbf{y}_j\|^2 \quad \text{and} \quad \frac{1}{n}\sum_j d_{ij}^2 = \|\mathbf{y}_i\|^2 + \frac{1}{n}\sum_j \|\mathbf{y}_j\|^2$$

$$\text{and also} \quad \frac{1}{n^2}\sum_{i,j} d_{ij}^2 = \frac{1}{n}\sum_j \frac{1}{n}\sum_i d_{ij}^2 = \frac{1}{n}\sum_i \|\mathbf{y}_i\|^2 + \frac{1}{n}\sum_j \|\mathbf{y}_j\|^2$$

which yields then

$$d_{ij}^2 - \frac{1}{n}\sum_i d_{ij}^2 - \frac{1}{n}\sum_j d_{ij}^2 + \frac{1}{n^2}\sum_{i,j} d_{ij}^2 = -2\mathbf{y}_i^\top \mathbf{y}_j$$

and subsequently the result. For the converse, just note that

$$\mathbf{B} = \mathbf{U}\mathbf{L}\mathbf{U}^\top = \mathbf{U}\mathbf{L}^{1/2}\mathbf{L}^{1/2}\mathbf{U}^\top$$

yielding the desired $\mathbf{Y} = \mathbf{U}\mathbf{L}^{1/2}$

Note that $\mathbf{B} = \mathbf{H}\mathbf{A}\mathbf{H}$, where $\mathbf{H} = \left(\mathrm{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top\right)$

## Low-rank approximation of the inner products

Next step starts with the matrix $\mathbf{B}$: let $\mathbf{U}\mathbf{L}\mathbf{U}^\top$ be the eigenvalue decomposition of $\mathbf{B}$

If the distances are Euclidean, then $\mathbf{B}$ is nonnegative definite, its eigenvalues are all nonnegative, and one can construct its approximation of rank $k$, a symmetric nonnegative definite matrix $\mathbf{B}_k = \mathbf{U}\mathbf{L}_k\mathbf{U}^\top$, where $\mathbf{L}_k$ is constructed from $\mathbf{L}$ by retaining only $k$ largest eigenvalues (and changing the rest of the diagonal to zero)

The eigenvalue decomposition $\mathbf{U}\mathbf{L}_k\mathbf{U}^\top$ of $\mathbf{B}_k$ then gives the coordinates of the new objects in the desired $k$-dimensional representation: $\mathbf{Y}_k = \mathbf{U}\mathbf{L}_k^{1/2}$

## Dirty tricks

All the above goes under the assumption that the dissimilarities are *Euclidean*; however, classical (metric) multidimensional scaling is often applied to arbitrary distance matrices, with no guarantee that they satisfy this assumption

The practical approach works under the tenet that the Euclidean assumptiom is satisfied *"at least approximately"*: that the dissimilarities arise, *possibly with some error*, as the Euclidean distances among *some* points in *some* Euclidean space

Out of these dissimilarities, the matrices **A** and **B** are constructed as above; this matrices are still symmetric, as long as the dissimilarities are symmetric - but may not be nonnegative definite: some eigenvalues may be negative

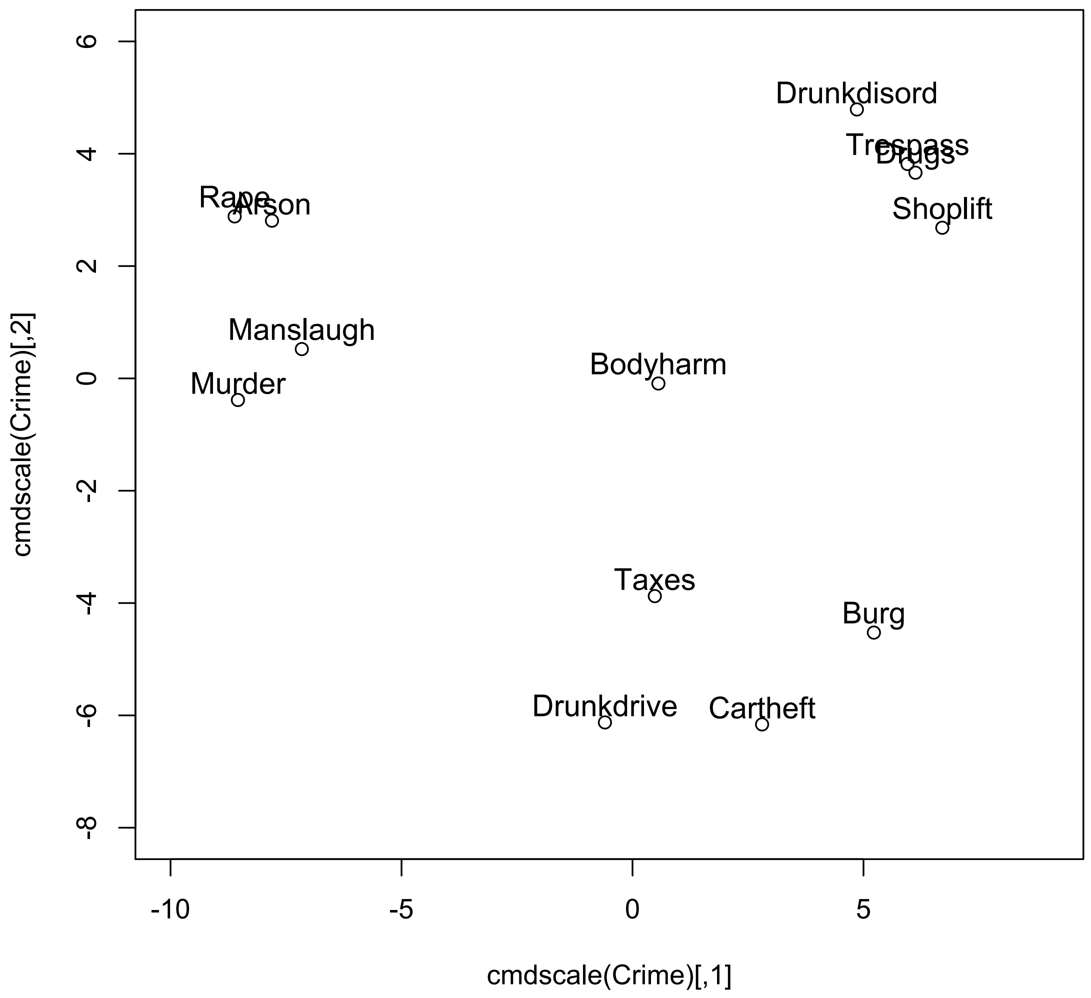
A possibility then is to increase all eigenvalues by a constant making them nonnegative; another possibility is to use only the $k$ largest eigenvalues, if they are positive

## Duality to principal components

If the dissimilarities $d_{ij}$ arise as the Euclidean distances between the lines of a matrix **Y** - which can be, due to the translation and rotation invariance of the distances, considered without loss of generality with centered columns, that is, the columnwise sums of **Y** are all zero - then the $k$-dimensional transformation amounts to obtaining the first $k$ principal components of **Y** (obtained without scaling to the correlation matrix)

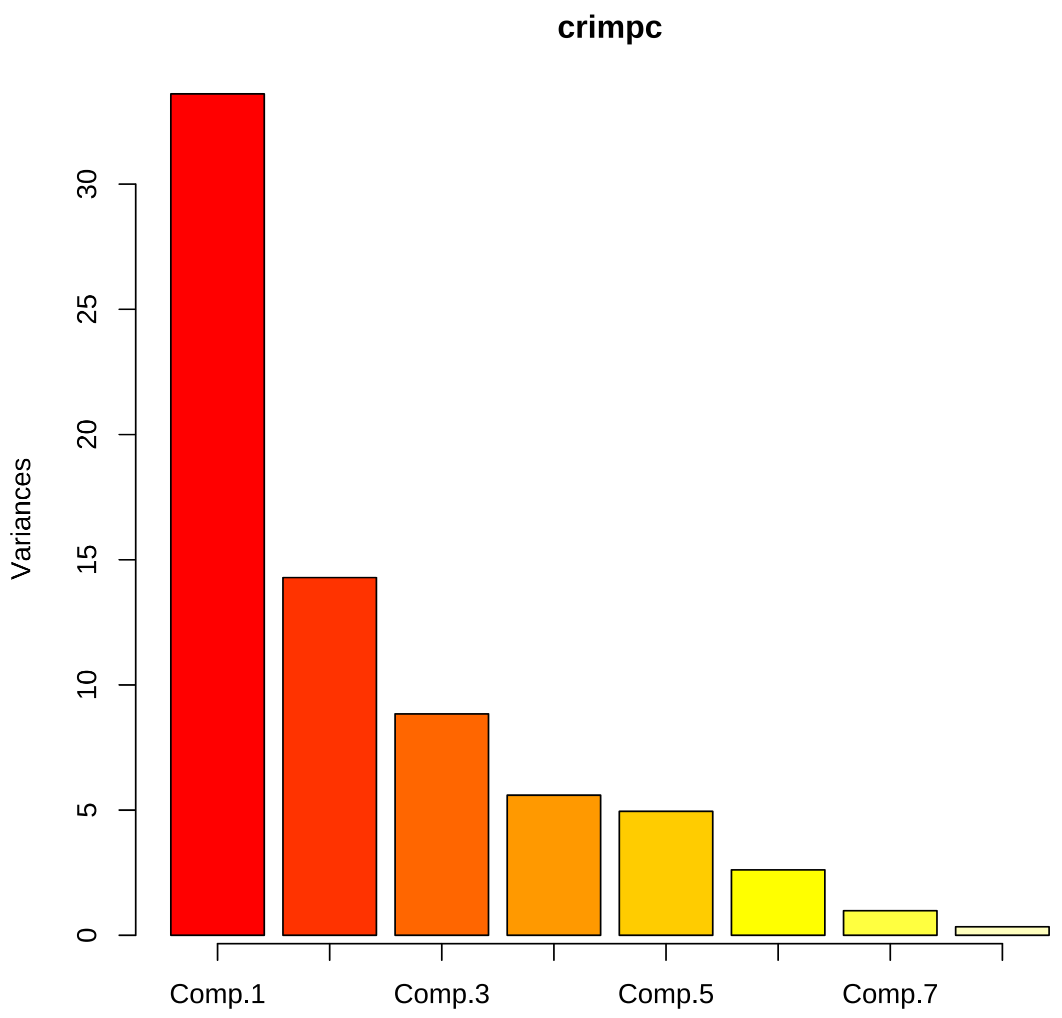
The classical multidimensional scaling can be thus seen as equivalent (or also dual) to obtaining first $k$ (most often $k = 2$, in which this amounts to plotting) principal components

# Classical multidimensional scaling



```
> plot(cmdscale(Crime),xlim=c(-10,9),ylim=c(-8,6))
> text(t(t(cmdscale(Crime))+c(0,0.3)),labels=labels(Crime),cex=1.1)
```

## How distorted?



0.6725267 % of variance explained by the first two components

## Best representation of dissimilarities?

At the beginning, we stipulated that multidimensional scaling aims at “representing data in a low (usually 2-) dimensional (hyper)plane while preserving the dissimilarities (distances) between the data points as much as possible (minimizing the possible distortion caused by the reduction of dimensionality)”.

Really? In fact, the just described procedure amounts to the best low-rank approximation, in the Euclidean metric, of the matrix **B** (with elements $b_{ij}$) representing (“reconstructing”) the original inner products: the new objects $\hat{\mathbf{y}}_i$, those in the lower-dimensional representation, thus minimize

$$\sum_{i,j} (b_{ij} - \hat{\mathbf{y}}_i^\top \hat{\mathbf{y}}_j)^2$$

among all possible representations

## Stress (or also STRESS)

How about the distances themselves? For instance, is it true that the result of classical multidimensional scaling minimizes something like

$$\sum_{i,j} \left(d_{ij} - \hat{d}_{ij}\right)^2$$

or a similar quantity?

Such a quantity is in this context called **stress**

(or also STRESS as an acronym, according to Kruskal: STandardized Residual Sum of Squares)

The particular stress formula is called *Kruskal-Shepard* stress

## The partial result for the classical scaling

If the distances are Euclidean, then the new distances, the distances between the new objects constructed by the algorithm of classical multidimensional scaling, are the ones minimizing, *among all linear transformations of the original objects*,

$$\sum_{i,j} (d_{ij}^2 - \hat{d}_{ij}^2) = \sum_{i \neq j} (d_{ij}^2 - \hat{d}_{ij}^2)$$

or equivalently

$$\frac{\sum_{i,j} (d_{ij}^2 - \hat{d}_{ij}^2)}{\sum_{i,j} d_{ij}^2} = \frac{\sum_{i \neq j} (d_{ij}^2 - \hat{d}_{ij}^2)}{\sum_{i \neq j} d_{ij}^2}$$

where $d_{ij}$ are the original and $\hat{d}_{ij}$ the new, transformed distances

The quantities above can be considered as a measure of distortion - **stress**

Non-metric (and, as a rule, nonlinear) versions of multidimensional scaling optimize (numerically) different stresses

## Sammon mapping

**Non-metric** (nonlinear) versions of **multidimensional scaling** construct the $k$-dimensional configuration via numerical minimization of a suitable stress function - *among all possible configurations, not just those obtained as linear projections*

The minimized stress in the multidimensional scaling method called **Sammon mapping** is (note that no off-diagonal zero elements allowed)

$$\frac{1}{\sum_{i \neq j} d_{ij}} \sum_{i \neq j} \frac{(d_{ij} - \hat{d}_{ij})^2}{d_{ij}}$$

The Sammon mapping stress gives more weight, more attention to small distances - rather than to large ones (based on the empirical finding that the large distances tend to be less precise than the small ones)

## Kruskal non-metric multidimensional scaling

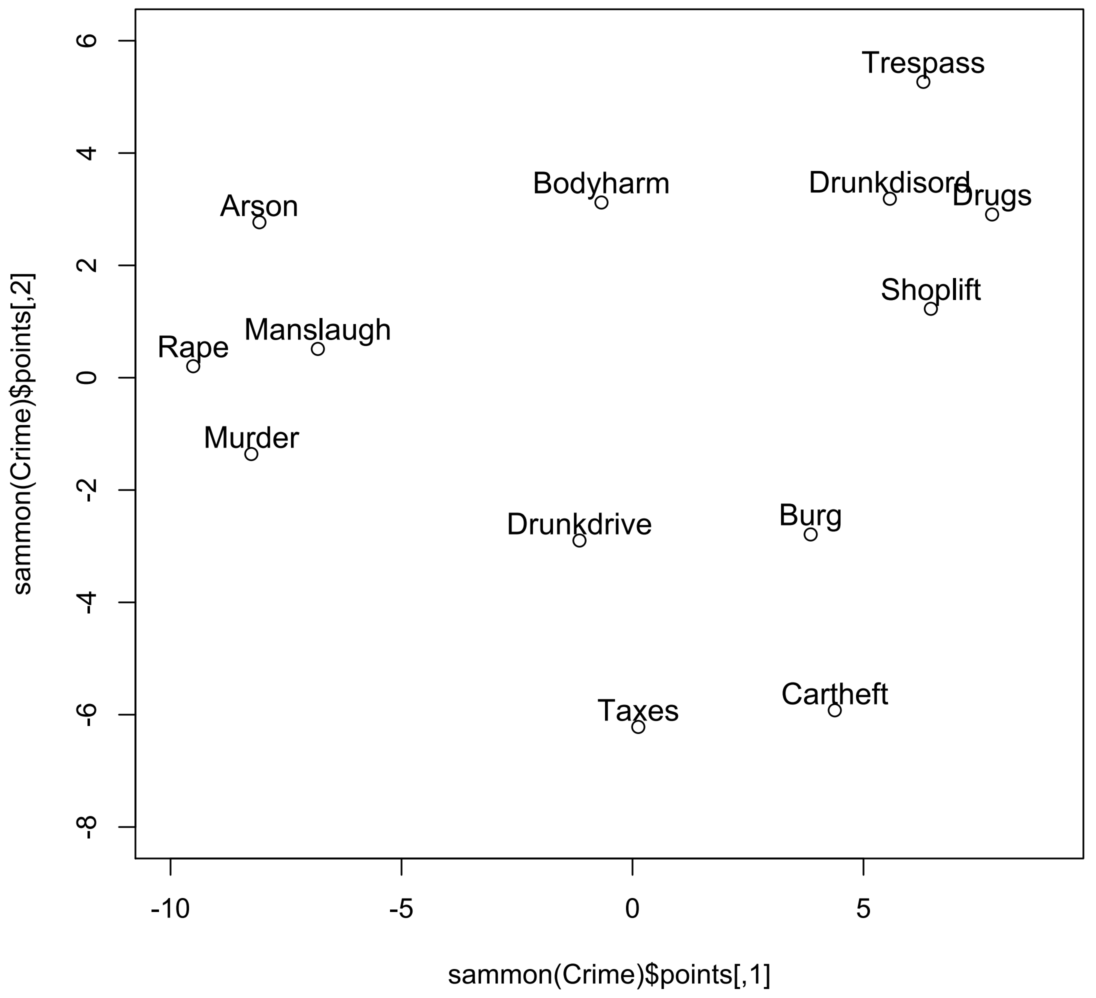
The stress of the **Kruskal nonmetric multidimensional scaling** (**iso**-MDS): is a modification of the Kruskal-Shepard stress

$$\sum_{i \neq j} \frac{(\vartheta(d_{ij}) - \hat{d}_{ij})^2}{\sum_{i,j} d_{ij}^2}$$

allowing also for an arbitrary increasing transformation $\vartheta$ - which means that the result does not depend that much on the original distances $d_{ij}$ but rather on their *ranks* (the ordinal numbers in their ordering); the idea is to choose a new configuration so that the *ordering* of the distances is as much preserved as possible

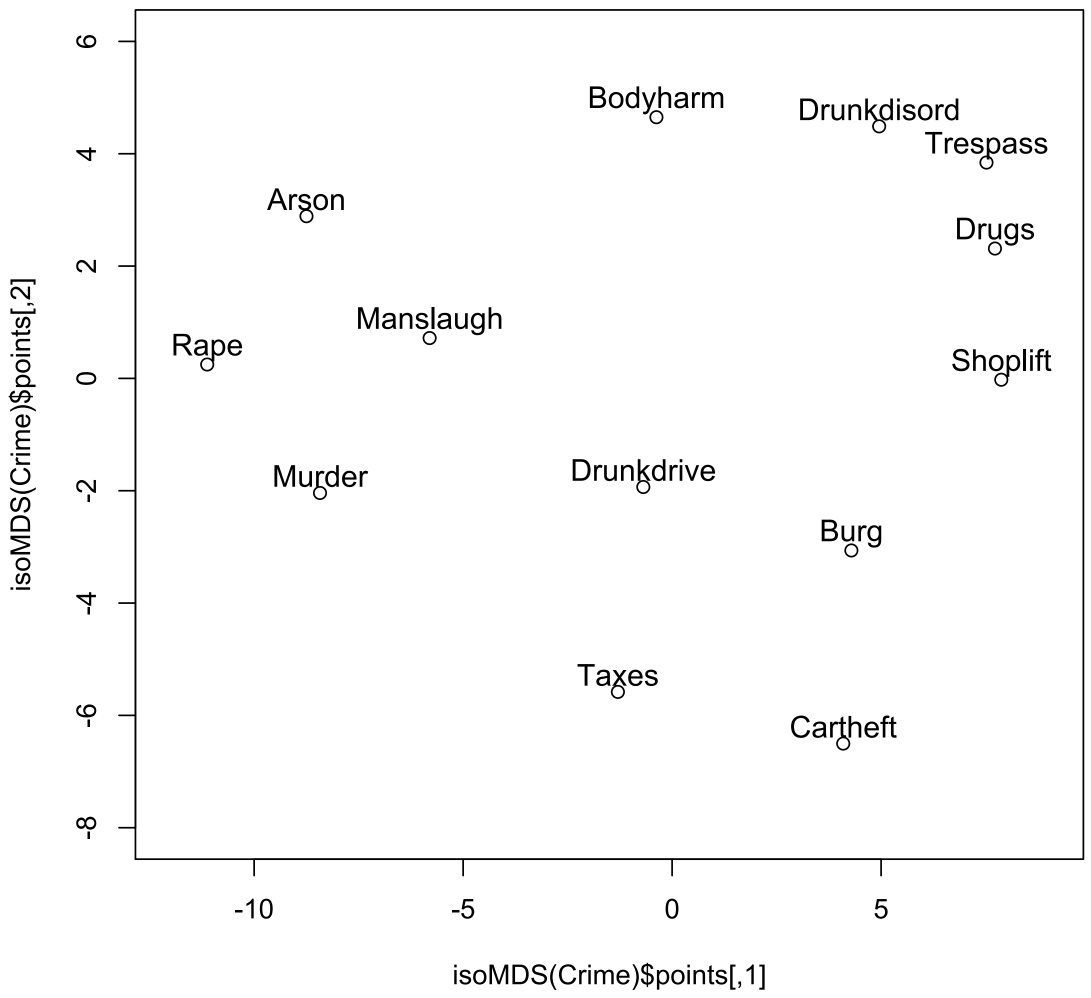
The algorithm minimizes over all possible $\hat{d}_{ij}$ as well as $\vartheta$.

## Sammon



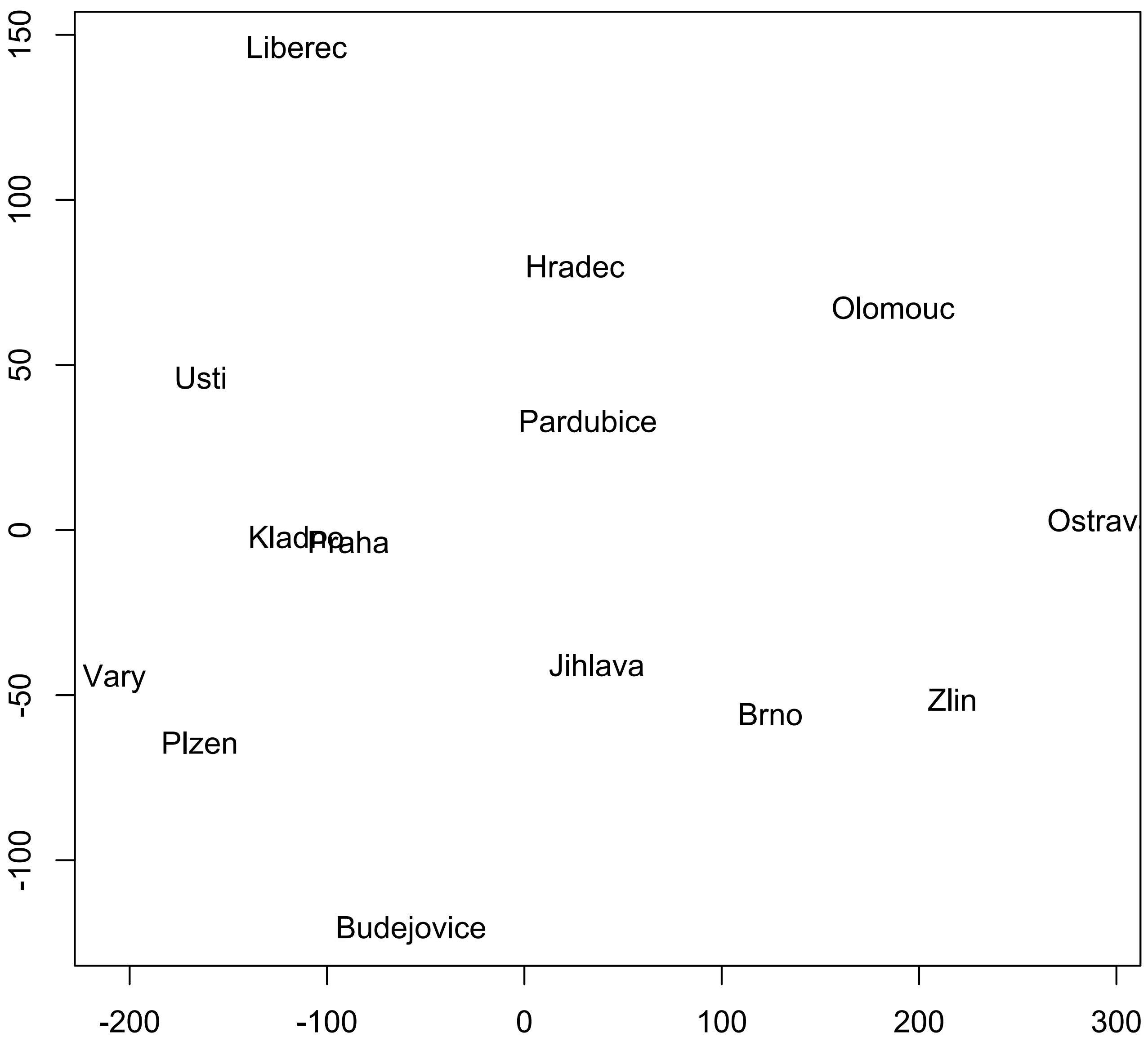
```
> library(MASS)
> plot(sammon(Crime)$points,xlim=c(-10,9),ylim=c(-8,6))
> text(t(t(sammon(Crime)$points)+c(0,0.3)),labels=labels(Crime),cex=1.1)
```

# Kruskal nonmetric



```
> library(MASS)
> plot(isoMDS(Crime)$points,xlim=c(-12,9),ylim=c(-8,6))
> text(t(t(isoMDS(Crime)$points)+c(0,0.3)),labels=labels(Crime),cex=1.1)
```

## And Czech cities? Not a big deal (Nic moc)



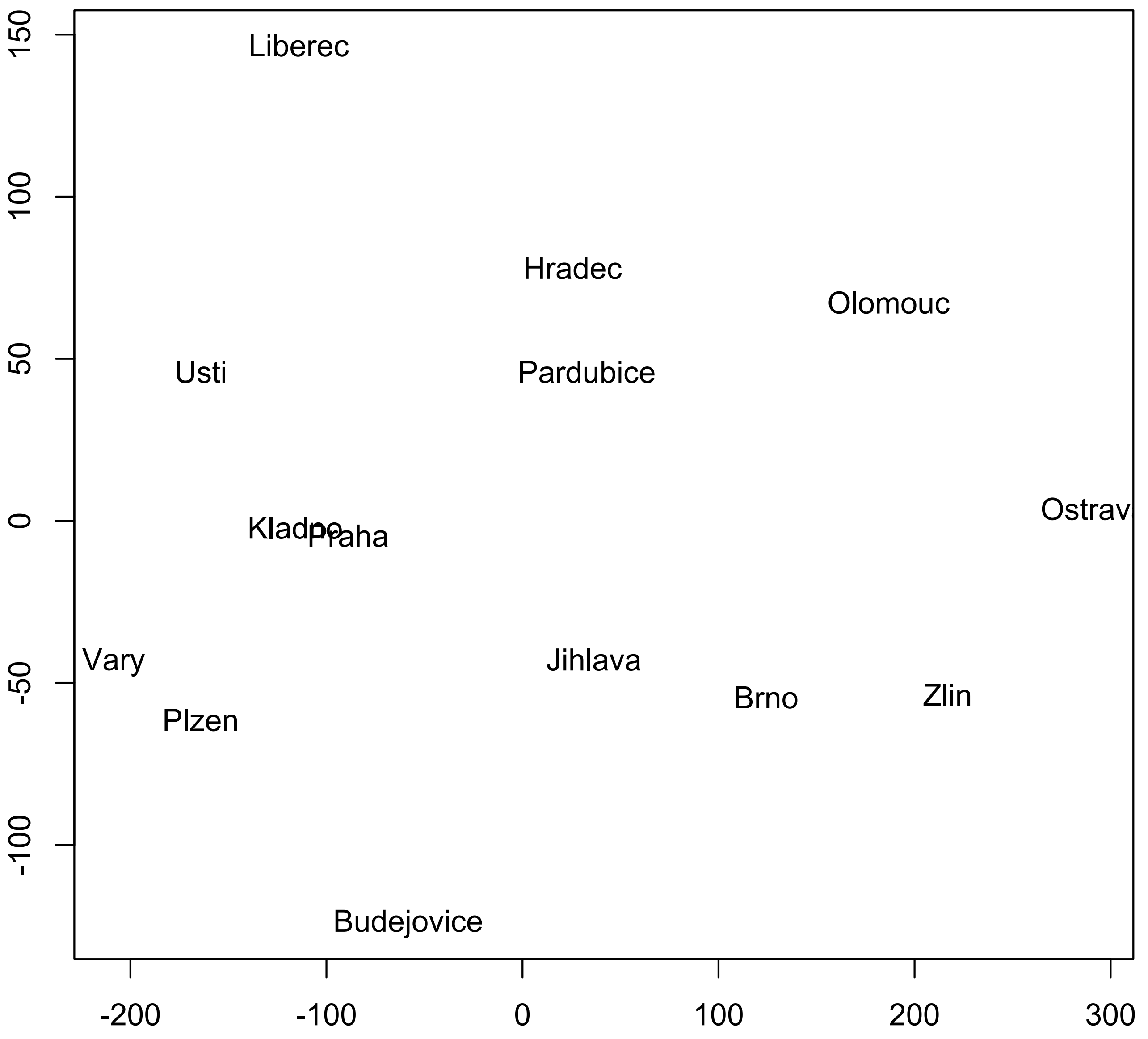
## Better data?

More thorough examination shows that the distance matrix is not symmetric... (???)

```
> Czechia=(Czechiaorg+t(Czechiaorg))/2
> Czech.cl=cmdscale(Czechia)
> plot(Czech.cl,type="n",xlab="",ylab="")
> text(Czech.cl,labels=rownames(Czech.cl))
```

```
           Brno Plzen Olomouc Liberec Budejovice Hradec  Usti Pardubice  Zlin Ostrava  Vary Jihlava Praha Kladno
Brno         0.0 296.0    77.0   308.0      216.5  167.5 297.0     146.0  96.0   169.5 337.0    93.0 208.5  240.0
Plzen      296.0   0.0   369.5   199.0      137.0  208.5 181.0     213.5 388.5   462.5  81.5   217.5  91.5   99.5
Olomouc     77.0 369.5     0.0   236.5      290.0  139.5 370.5     135.0  62.5    96.0 410.5   167.0 282.0  313.5
Liberec    308.0 199.0   236.5     0.0      249.5   97.5  96.0     154.5 400.5   437.0 233.5   229.5 109.5  135.0
Budejovice 216.5 137.0   290.0   249.5        0.0  253.0 238.5     225.5 308.5   383.0 218.0   138.0 150.5  182.0
Hradec     167.5 208.5   139.5    97.5      253.0    0.0 190.0      24.0 206.0   238.5 240.5   112.5 116.0  143.0
Usti       297.0 181.0   370.5    96.0      238.5  190.0   0.0     196.0 389.5   463.0 121.5   218.5  92.5   86.5
Pardubice  146.0 213.5   135.0   154.5      225.5   24.0 196.0       0.0 201.5   234.5 246.0    90.0 121.0  148.5
Zlin        96.0 388.5    62.5   400.5      308.5  206.0 389.5     201.5   0.0   124.5 429.5   186.0 300.5  332.5
Ostrava    169.5 462.5    96.0   437.0      383.0  238.5 463.0     234.5 124.5     0.0 503.0   260.0 375.0  406.5
Vary       337.0  81.5   410.5   233.5      218.0  240.5 121.5     246.0 429.5   503.0   0.0   258.5 127.5  105.0
Jihlava     93.0 217.5   167.0   229.5      138.0  112.5 218.5      90.0 186.0   260.0 258.5     0.0 130.0  161.5
Praha      208.5  91.5   282.0   109.5      150.5  116.0  92.5     121.0 300.5   375.0 127.5   130.0   0.0   31.0
Kladno     240.0  99.5   313.5   135.0      182.0  143.0  86.5     148.5 332.5   406.5 105.0   161.5  31.0    0.0
```

# Not a big deal either



## Hm...

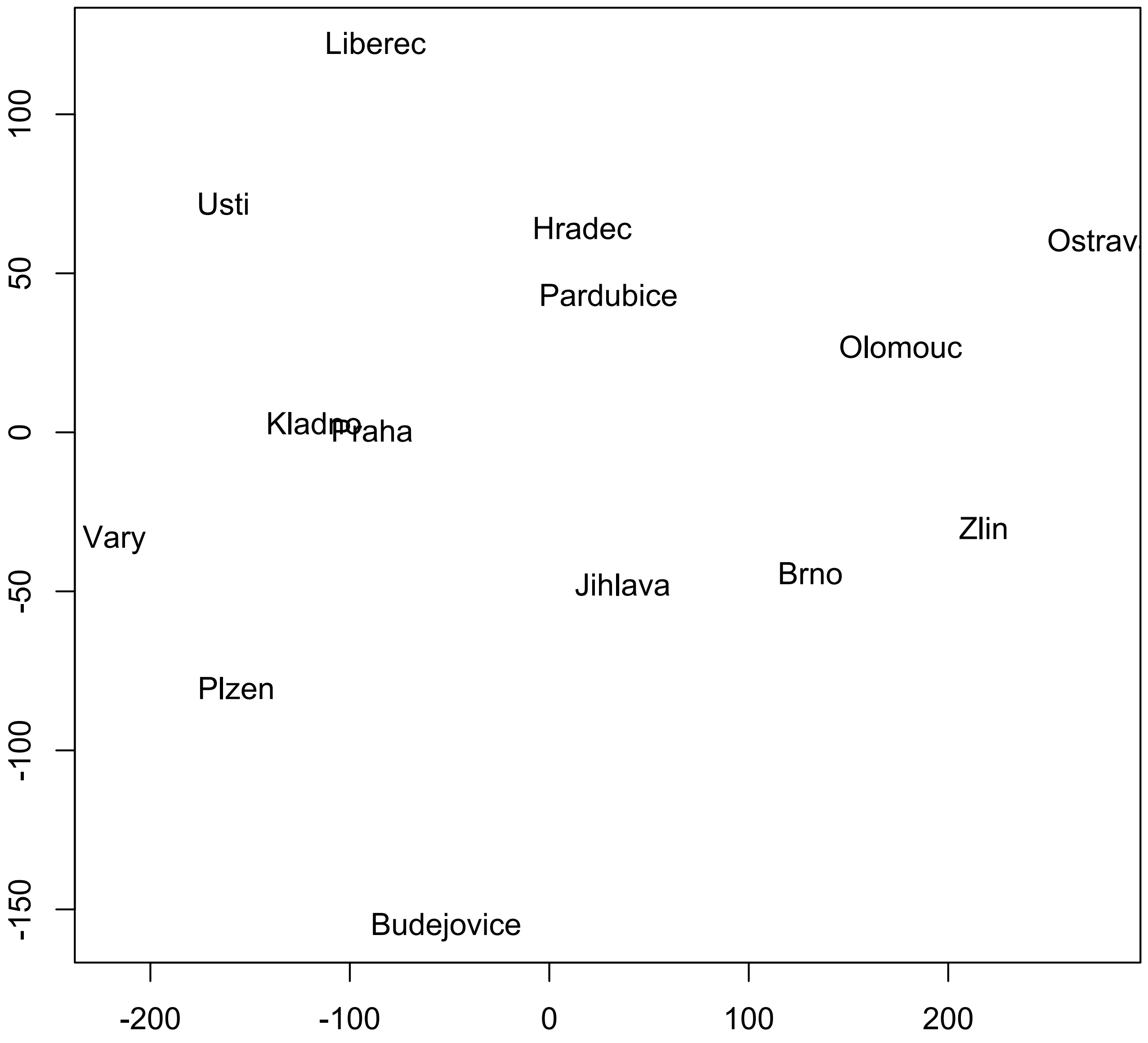
That Ostrava is maybe really far away from everything...

```
      Brno       Plzen    Olomouc    Liberec Budejovice
    2652.0      2945.5     2949.5     2886.0     2990.0
    Hradec        Usti  Pardubice       Zlin    Ostrava
    2136.5      2940.5     2136.0     3426.0     4153.0
      Vary     Jihlava      Praha     Kladno
    3312.0      2262.0     2135.5     2384.5
```

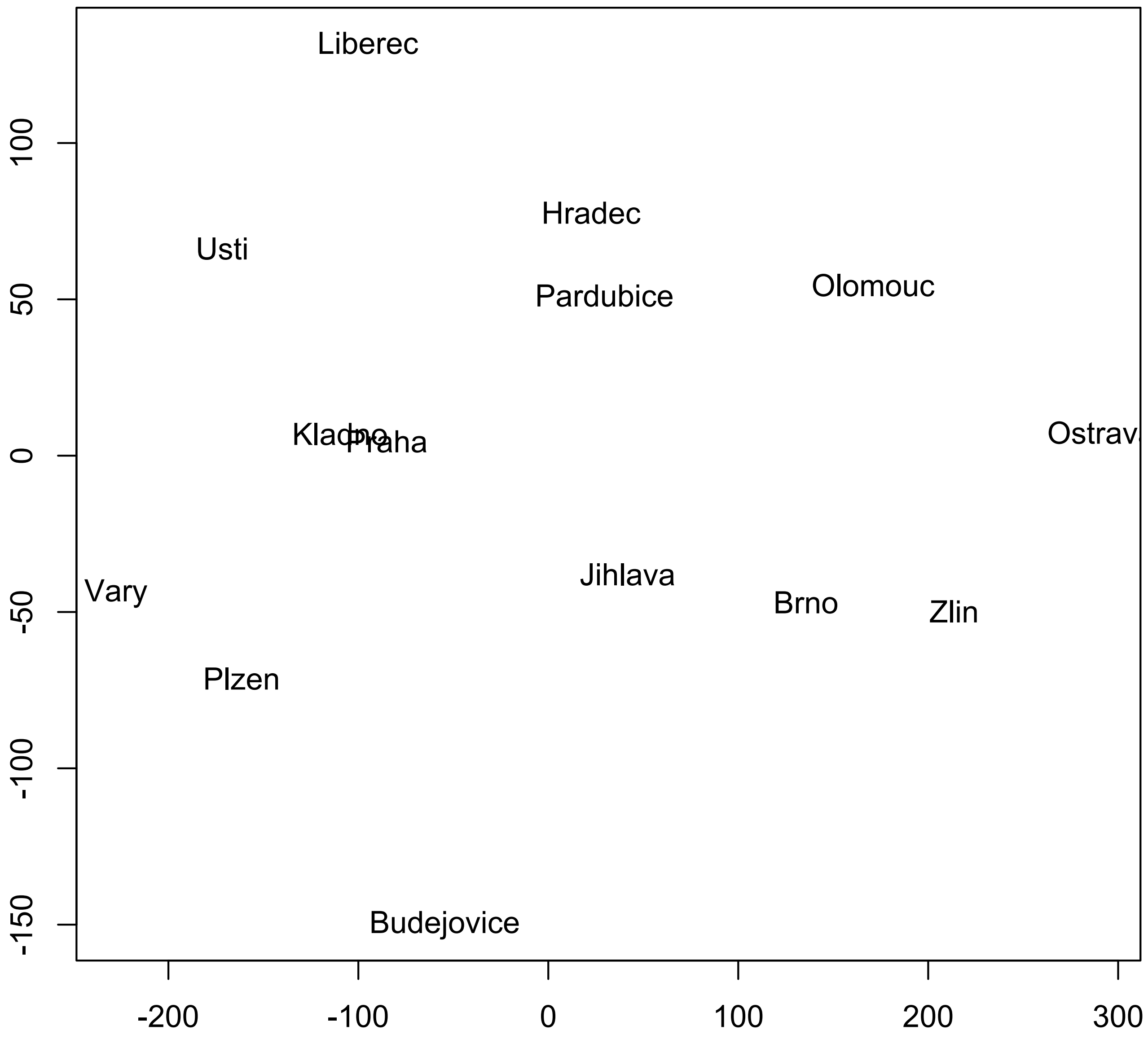
So how about...

```
> Czech.sam=sammon(Czechia)
Initial stress        : 0.01890
stress after  10 iters: 0.00528, magic = 0.500
stress after  20 iters: 0.00525, magic = 0.500
> plot(Czech.sam$points,type="n",xlab="",ylab="")
> text(Czech.sam$points,labels=rownames(Czech.sam$points))
```

## Bingo? Perhaps after a bit of rotation...



## ...but not with Kruskal



## Graves with pottery

In an archaelogical study there are six graves containing five types of pottery between them

```
> Graves
  1 2 3 4 5
A 0 0 1 1 0
B 1 1 0 0 1
C 0 1 1 1 1
D 0 0 1 1 0
E 1 0 0 0 1
F 1 0 1 1 1
```

It is desired to order the graves chronologically based on the similarity/dissimilarity of the pottery found in them.

We are going to look at the dissimilarities of the rows, $x_i$, of the matrix shown above; note that the distance of lines A and D will be inevitably 0 in any reasonable case

## Relevant dissimilarities

The Hamming distance is in this case equal to the $d_1(x_j, x_j)/5$, the $\ell^1$ ("Manhattan") distance divided by $p = 5$; the following shows the result for the $\ell^1$ distance

```
> dist(Graves,method="manhattan",diag=TRUE)
  A B C D E F
A 0
B 5 0
C 2 3 0
D 0 5 2 0
E 4 1 4 4 0
F 2 3 2 2 2 0
```

## Transforming similarities to dissimilarities

The book of Mardia, Kent, and Bibby champions using an ordinary matching coefficient $c(x_i, x_j)$, which counts the number of cases when both vectors are either equal to 0 or to 1, and divides them by $p = 5$: the result is

$$c(x_i, x_j) = 1 - d_1(x_i, x_j)/5.$$

In terms of the book of Mardia, Kent, and Bibby, $c(x_i, x_j)$ is a "similarity" which they convert into an euclidean dissimilarity by the transformation

$$d(x_i, x_j) = (c(x_i, x_i) - 2c(x_i, x_j) + c(x_j, x_j))^{1/2}$$

which in the present case yields

$$d(x_i, x_j) = (2 - 2c(x_i, x_j))^{1/2} = \sqrt{2d_1(x_i, x_j)/5}$$

```
> graveu <- sqrt(2*dist(Graves,method="manhattan",diag=TRUE)/5)
> eigen((diag(6)-1/6) %*% as.matrix(-graveu^2/2) %*% (diag(6)-1/6))
eigen() decomposition
$values
[1]  1.747708e+00  5.864466e-01  3.518536e-01
[3]  4.732485e-02  0.000000e+00 -9.685690e-17
...
```

## Seriation: scaling into dimension one

We perform now classical multidimensional scaling into dimension $k = 1$, to obtain a possible time order of the graves

```
> gramds <- cmdscale(graveu,k=1)
> gramds
          [,1]
A  0.600019876
B -0.765783233
C  0.194649330
D  0.600019876
E -0.635067161
F  0.006161312
```

This suggests an order (A and D), C, F, E, B - or the reverse one; note that time orientation cannot be determined from the method

## Seriation without euclidization

Of course, we could apply the algorithm directly to the $\ell^1$ distance - as we are scaling into dimension one, we need just the maximal eigenvalue to be positive

```
> cmdscale(dist(Graves,method="manhattan",diag=TRUE)/5,k=1)
          [,1]
A  0.414792257
B -0.566671301
C  0.136621560
D  0.414792257
E -0.408900852
F  0.009366079
> cmdscale(dist(Graves,method="manhattan",diag=TRUE),k=1)
        [,1]
A  2.0739613
B -2.8333565
C  0.6831078
D  2.0739613
E -2.0445043
F  0.0468304
```

The resulting orders are the same: (A and D), C, F, E, B

## Something else

Finally, note that the application of the nonlinear metholds is hampered by the fact that lines A and D have zero distance. Nonetheless, after omitting the line A, we obtain

```
> sammon(dist(Graves[-1,],method="manhattan",diag=TRUE)/5,k=1)
...
         [,1]
B  0.49885561
C -0.27272060
D -0.53060611
E  0.34010348
F -0.03563238
```

which yields the same order: D (and A), C, F, E, B

## Something else

Only the Kruskal method yields somewhat different result - but note that the scores are very close there, suggesting that while the groups (A, D, C, F) and (B, E) may be really apart, the ordering within the groups may be a bit quastionable

```
> isoMDS(dist(Graves[-1,],method="manhattan",diag=TRUE)/5,k=1)
...
        [,1]
B  0.5048145
C -0.3365613
D -0.3365814
E  0.5048155
F -0.3364872
...
```

## Multidimensional scaling: stochastic underpinning?

James O. Ramsay: Some Statistical Approaches to Multidimensional Scaling

Journal of the Royal Statistical Society. Series A (General), Vol. 145, No. 3 (1982), pp. 285-312

(with discussion)

# Finding Groups in Data: Cluster Analysis

## Two main approaches

Cluster analysis finds groups in the data - groups are formed by those items (datapoints) that have small distances (dissimilarities)

Dissimilarities - we have seen many of them, some are metrics ($d_{ij} \leqslant d_{ik} + d_{kj}$), some are more general; hierarchical clustering (in fact) involves *ultrametrics* ($d_{ij} \leqslant \max\{d_{ik}, d_{kj}\}$ - tree distance, see below).

The objective is to divide items into $k$ groups so that the members of the same group are more alike than members of the other group: *clusters*.

The number $k$ may be preassigned: **partitioning methods**

or not preassigned: **hierarchical clustering**.

A lot of clustering methods (a majority?) are based on distances between the items; so cluster analysis is often viewed as a distance-base method.

Methods: computational (rather than "statistical").

## R

Earlier packages:

`kmeans`: classical k-means routine, base R

`hclust`: some hierarchical clustering, but also still graphical routines in use, for instance more customizable plotting of dendrograms via `as.dendrogram(as.hclust())`)

`plclust`: *deprecated*, use `pltree` (or `plot`) from the package `cluster`

Benchmark:

`cluster`, functions `agnes` (AGglomerative NESting, agglomerative hierarchical clustering), `clara` (Clustering LARge Applications, partitioning method), `daisy` (Dissimilarity matrix calculation), `diana` (DIvisive ANAlysis clustering, divisive hierarchical clustering), `fanny`, (Fuzzy ANalysis clustering), `mona` (MONothetic clustering of binary variables, hierarchical clustering), `pam` (Partitioning Around Medoids, partitioning method)

Kaufman, L. and Rousseeuw, P. J. (1990): Finding Groups in Data: An Introduction to Cluster Analysis. Wiley, New York.

## Partitioning methods: k-means

The number of clusters, $k$, is given in advance. The clusters themselves are determined by minimizing a criterion.

**k-means**: minimize the sum of squared distances, within clusters, from the centers of clusters - which are subject to minimization too

The minimized criterion is the minimum

over the selection of clusters and over the selection of $\mathbf{m}_i$'s,

$$\sum_{i=1}^{k} \sum_{j \in C_i} \|\mathbf{x}_{ij} - \mathbf{m}_i\|^2$$

which is equal to the minimum over the selection of clusters, of

$$\sum_{i=1}^{k} \sum_{j \in C_i} \|\mathbf{x}_{ij} - \bar{\mathbf{x}}_i\|^2$$

where $\mathbf{x}_i$ is the mean of $i$-th cluster $C_i$.

We are looking for that partition to clusters $C_1, C_2, \ldots, C_k$ that makes these sums minimal

## Partitioning methods: k-medoids

The number of clusters, $k$, is given in advance. The cluster themselves are determined by minimizing a criterion.

**k-medoids**: the sum of distances, within clusters, from the centers of clusters - which are subject of minimization too; here it is also required that the center of the group is a data point itself (then general dissimilarities may be used)

The minimized criterion is the minimum, over the selection of clusters and over the selection of $\mathbf{m}_i$'s,

$$\sum_{i=1}^{k} \sum_{j \in C_i} \|\mathbf{x}_{ij} - \mathbf{m}_i\|$$

In analogy to k-means, we could take $\mathbf{m}_i$'s to be the spatial medians of the $i$-th cluster - we only have the additional requirement that *they should be one of the datapoints*, which makes computation easier

## Let us try it

```
> kmea1=kmeans(trackmen,3)
> kmea1
...
Clustering vector:
argentin australi  austria  belgium  bermuda   brazil    burma   canada
       3        1        3        1        3        1        3        1
   chile    china columbia   cookis    costa    czech  denmark   domrep
       1        1        1        2        3        1        1        2
 finland   france      gdr      frg     gbni   greece guatemal  hungary
       1        1        1        1        1        1        3        1
   india indonesi  ireland   israel    italy    japan    kenya    korea
       1        2        1        3        1        1        1        3
dprkorea luxembou malaysia mauritiu   mexico netherla       nz   norway
       1        3        2        2        1        1        1        1
     png philippi   poland portugal  rumania singapor    spain   sweden
       2        3        1        1        1        2        1        1
switzerl   taipei thailand   turkey      usa     ussr   wsamoa
       1        3        2        1        1        1        2
...
```

Is it not just clustering along the marathon records?

## We could check it out

```
> kmea2=kmeans(trackmen[,8],3)
> kmea2
...
Clustering vector:
 [1] 1 3 1 3 1 3 1 3 3 3 3 2 1 3 3 2 3 3 3 3 3 3 1 3 3 2 3 1
[29] 3 3 3 1 3 1 2 2 3 3 3 3 2 1 3 3 3 2 3 3 3 1 2 3 3 3 2
...
> as.vector(kmea1$cluster)
 [1] 3 1 3 1 3 1 3 1 1 1 1 2 3 1 1 2 1 1 1 1 1 1 3 1 1 2 1 3
[29] 1 1 1 3 1 3 2 2 1 1 1 1 2 3 1 1 1 2 1 1 1 3 2 1 1 1 2
```

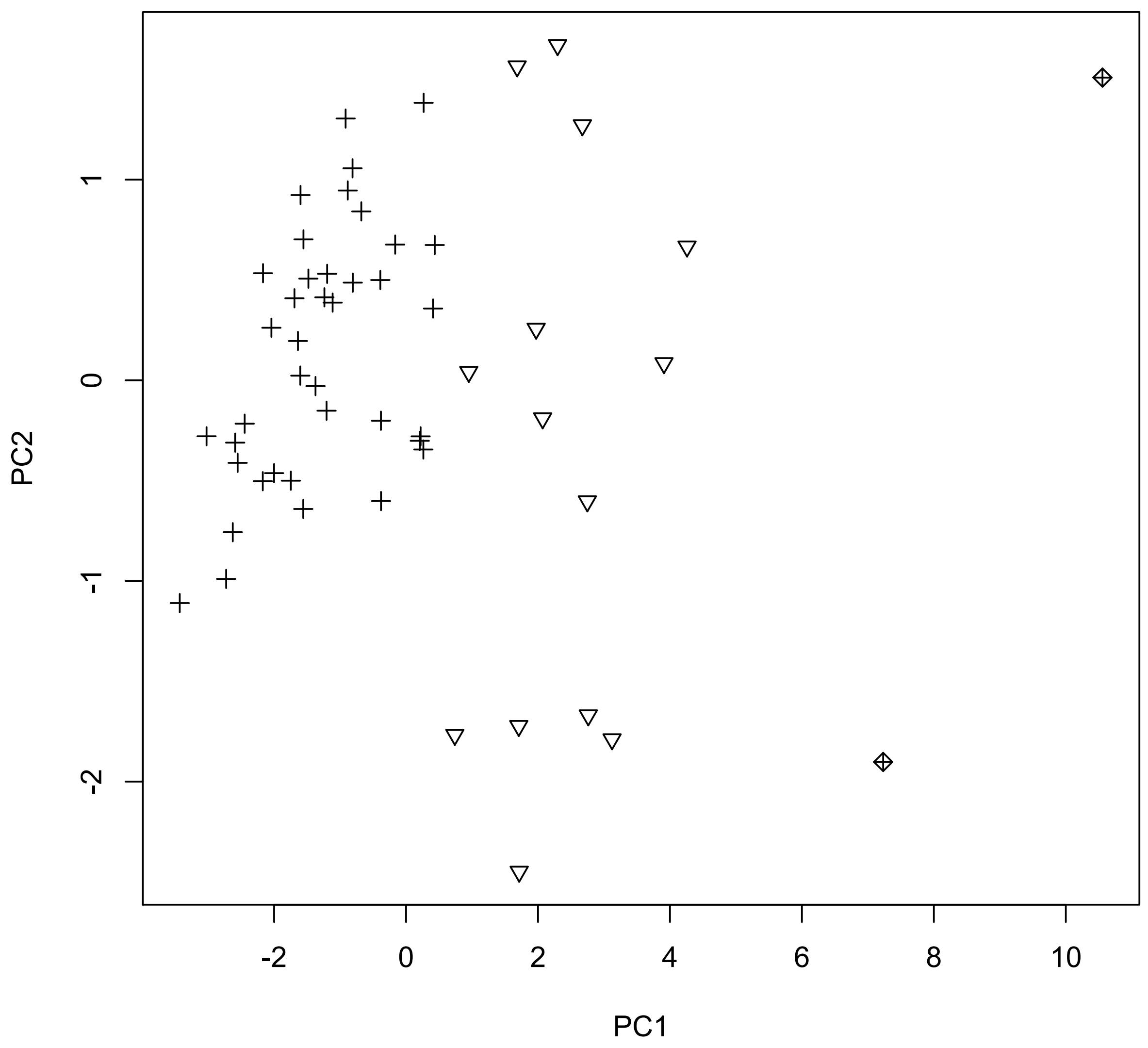
Are they not the same? Indeed

```
> c(3,2,1)[kmea2$cluster]-as.vector(kmea1$cluster)
 [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[29] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

## So, scaling once again

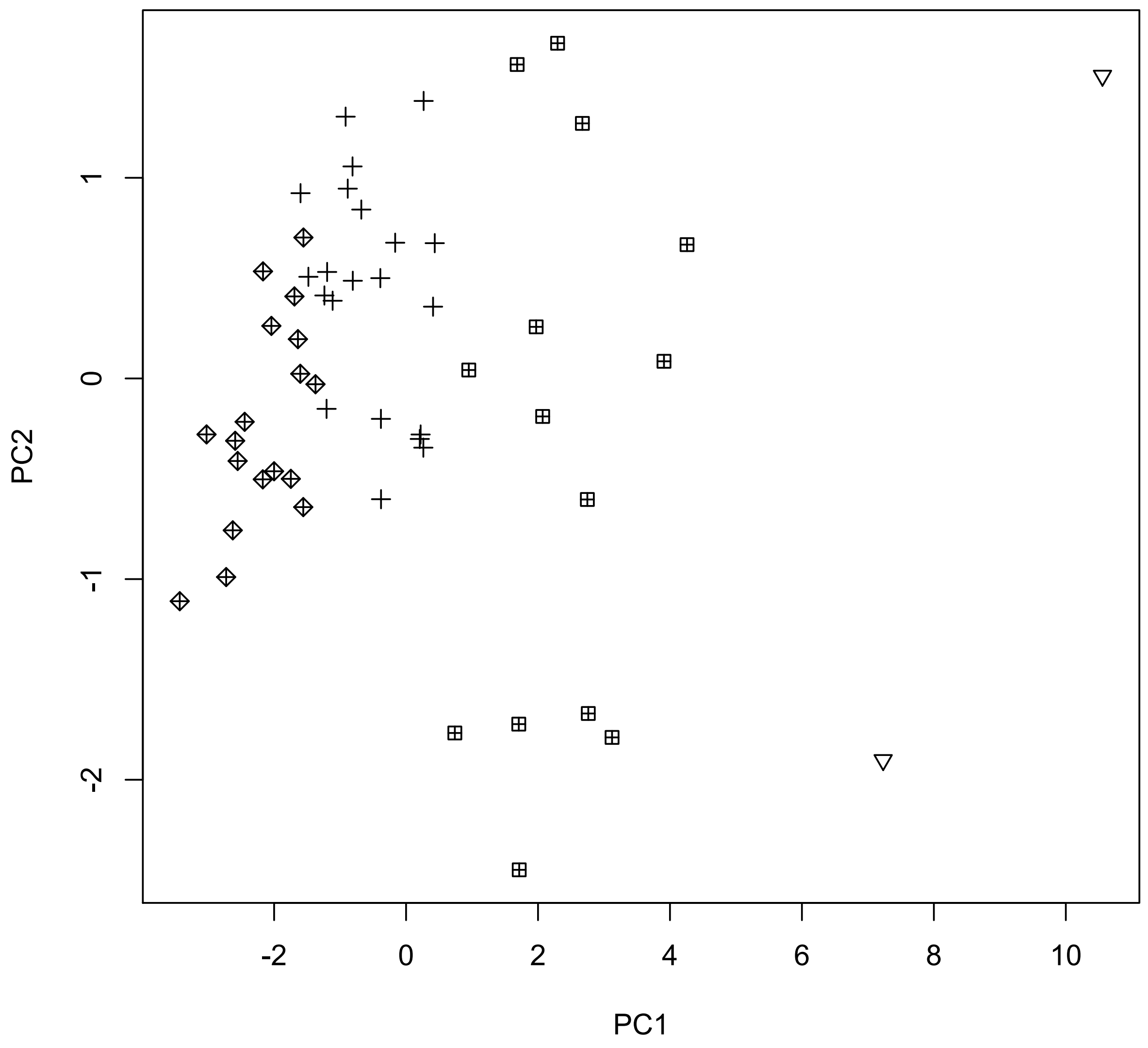
```
> kmea3=kmeans(scale(trackmen),3)
> kmea3
...
Clustering vector:
argentin australi  austria  belgium  bermuda   brazil    burma
       1        1        1        1        2        1        2
  canada    chile    china columbia   cookis    costa    czech
       1        1        1        1        3        2        1
 denmark   domrep  finland   france      gdr      frg     gbni
       1        2        1        1        1        1        1
  greece guatemal  hungary    india indonesi  ireland   israel
       1        2        1        1        2        1        1
   italy    japan    kenya    korea dprkorea luxembou malaysia
       1        1        1        1        2        1        2
mauritiu   mexico netherla       nz   norway      png philippi
       2        1        1        1        1        2        2
  poland portugal  rumania singapor    spain   sweden switzerl
       1        1        1        2        1        1        1
  taipei thailand   turkey      usa     ussr   wsamoa
       2        2        1        1        1        3
```

## A cautious picture



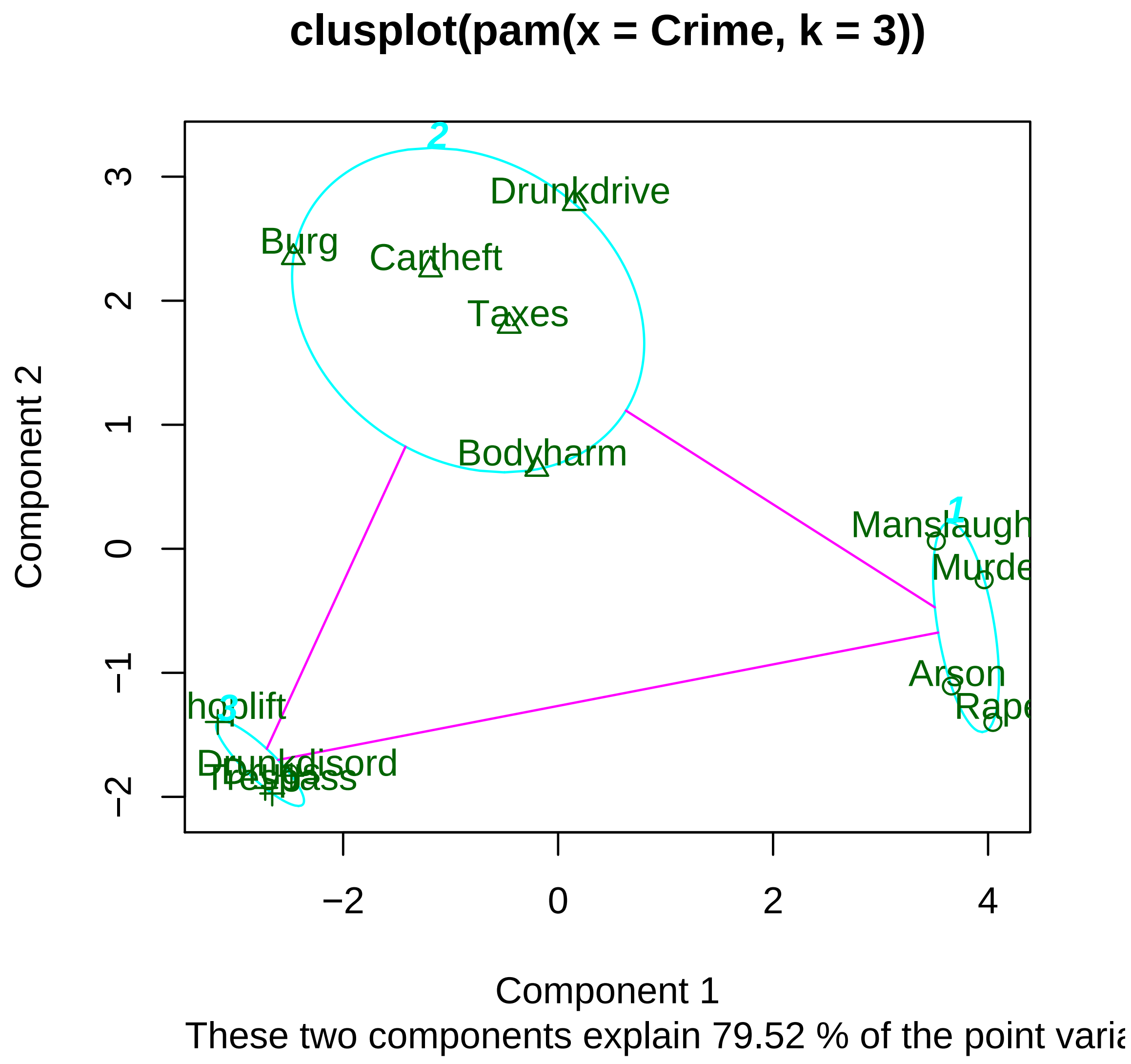
```
plot(predict(prcomp(trackmen,scale=T)),pch=3*kmea3$cluster)
```

# 4-means?

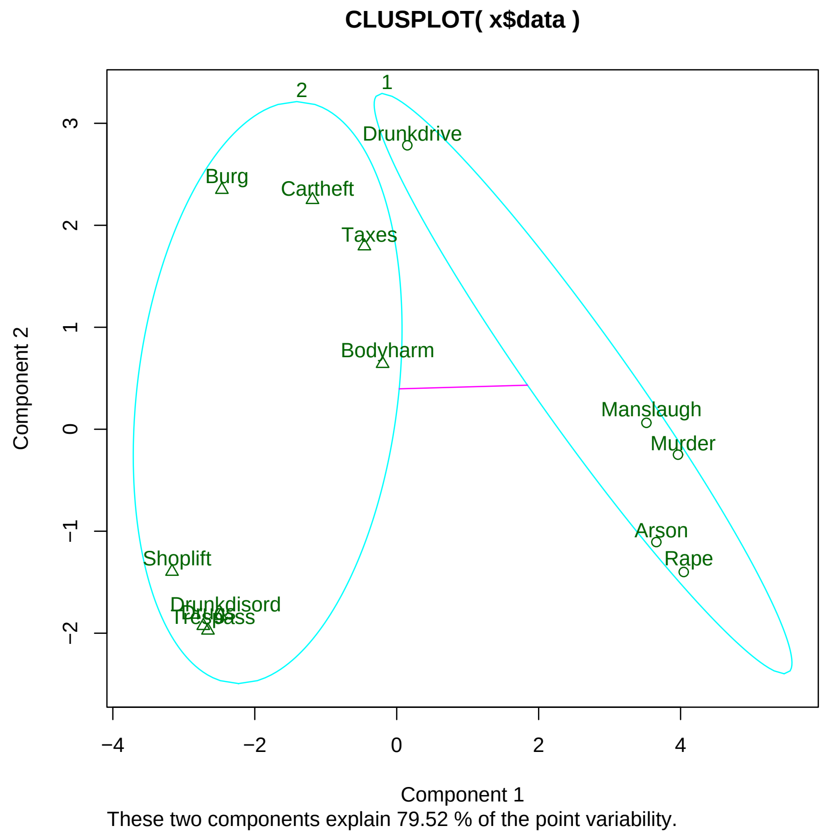


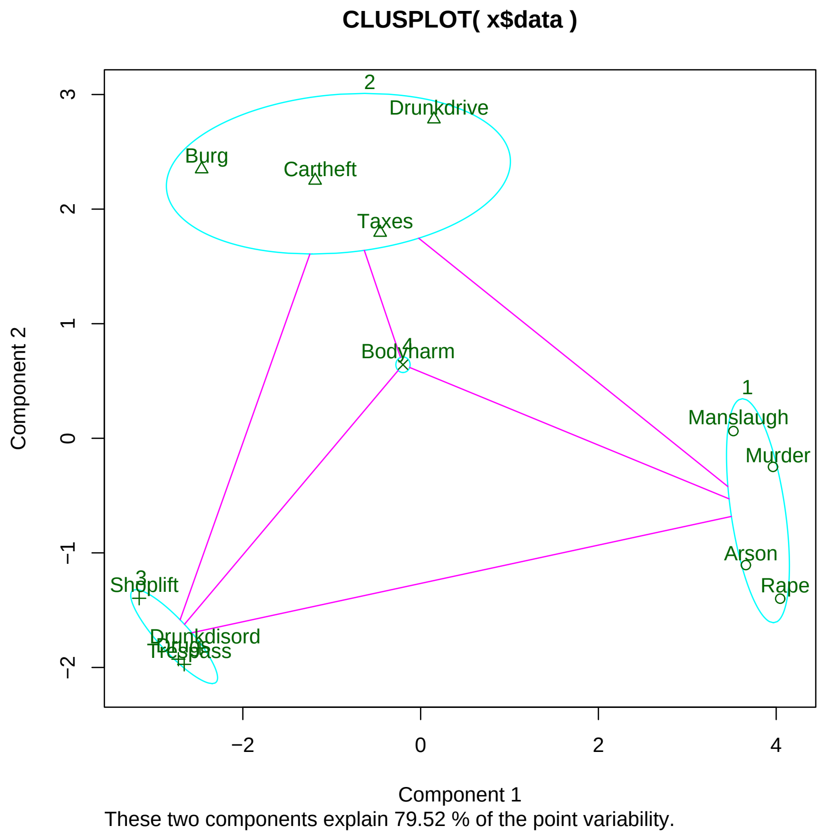
# Crimes: 3-medoids

```
> library(cluster)
> clusplot(pam(Crime,3),labels=2,sub='')
```



# Crimes: k-medoids with different $k$

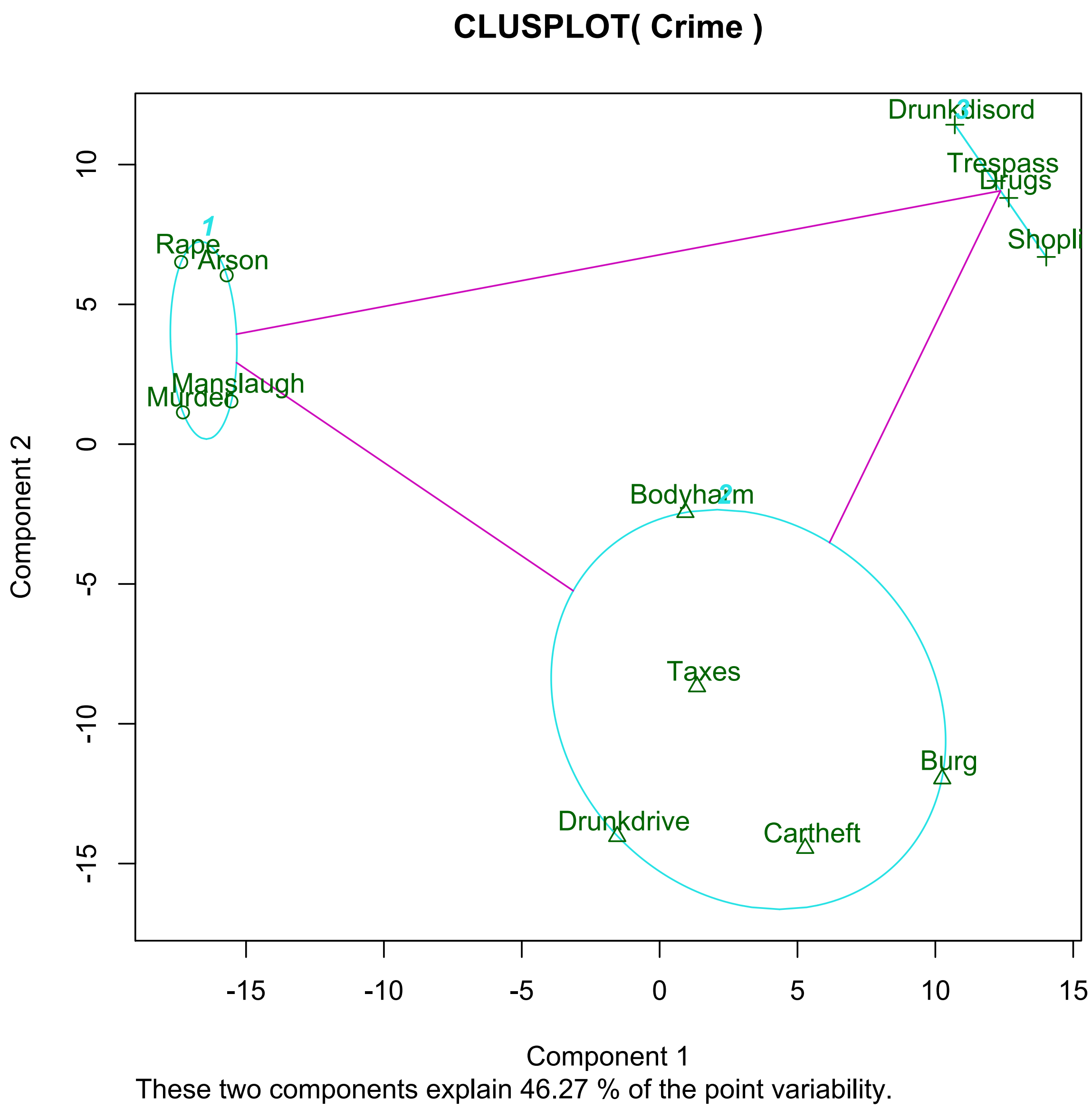




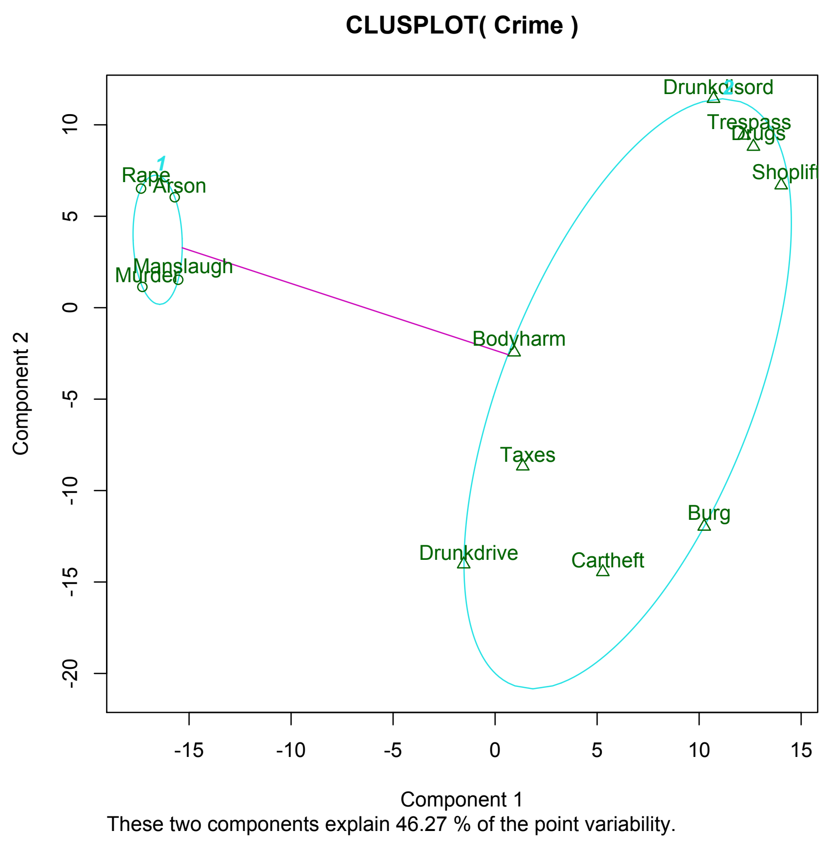
k-means give Cartheft to the other when $k = 2$;
$k = 4$: `Bodyharm` extra

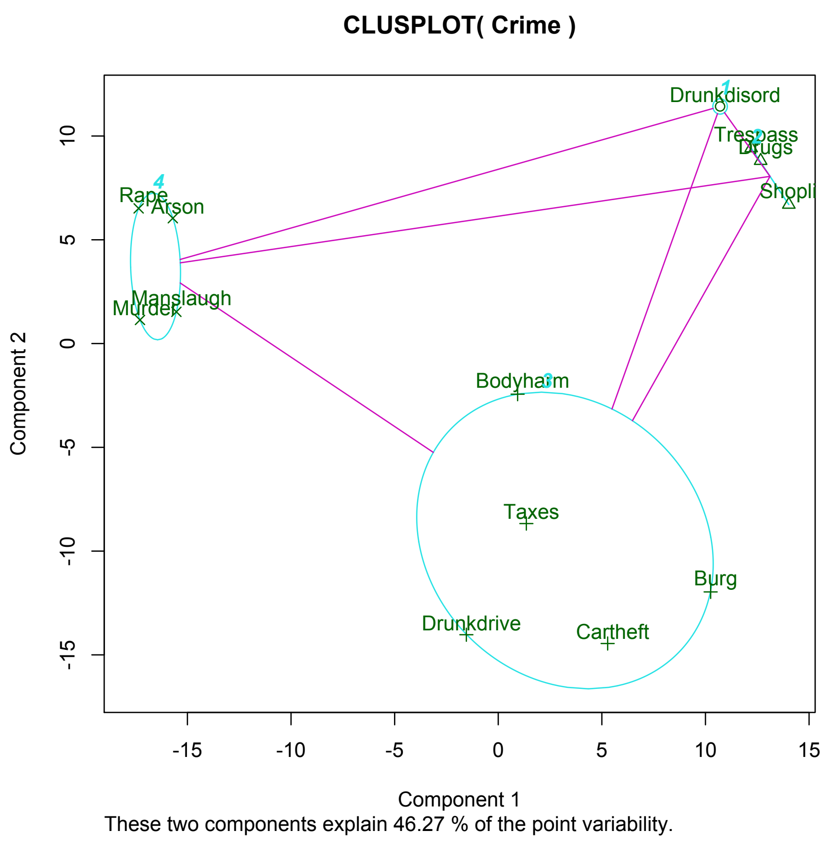
# Crimes: 3-means

```
> clusplot(Crime,kmeans(Crime,3)$cluster,diss=T,labels=2)
```



# Crimes: k-means with different $k$





$k = 2$: different from 2-medoids

$k = 4$: `Drunkdisord` is extra

## Hierarchical clustering

We may:

• start with clusters containing 1 item and end up with all items in one cluster: **agglomerative** clustering

• or other way round: divide and divide: **divisive** clustering

At every step, we are trying to join some two clusters (agglomerative) or divide some cluster (divisive). A criterion for that may be

• **single linkage**: clusters are close if each contains one element such that the two are close (*take min of distances*)

• **complete linkage**: clusters are close if each element of the first is close to each element of the second (*max of distances*)

• **average linkage**: clusters are close if the elements are close on average (each to each) (*average of distances*)

## Divisive methods

An example of divisive method is the algorithm of MacNaughton and Smith: the item is put into a new cluster if the average dissimilarity to the objects of that cluster is smaller than the average dissimilarity to the members of its old cluster.

At each stage, the cluster with the largest diameter is selected. (The diameter of a cluster is the largest dissimilarity between any two of its observations.) To divide the selected cluster, the algorithm first looks for its most disparate observation (i.e., which has the largest average dissimilarity to the other observations of the selected cluster). This observation initiates the ”splinter group”. In subsequent steps, the algorithm reassigns observations that are closer to the ”splinter group” than to the ”old party”. The result is a division of the selected cluster into two new clusters.

(Implemented in function `diana` from the package `cluster`.)

## Dendrogram

The specific plot for cluster analysis: **dendrogram** (in principle, can be used also for partitioning methods, we just vary $k$). From the graph theory perspective, it is a tree: a connected graph with no cycles

The dissimilarity that can be read from the dendrogram is the *tree distance* (of cases and clusters): the value of this distance is the maximum *height* we have to get to on the dendrogram when traveling from one cluster or object to another connecting the two clusters or objects, the value of the distance in the moment they enter the common cluster. This dissimilarity is an ultrametric

For two original objects, however, their tree distance is not necessarily equal to their original dissimilarity - this is true only when the original dissimilarity is itself an ultrametric, and also not true for every type of linkage. From this aspect, the dendrogram should be viewed rather as an approximation

## A Scotch Example

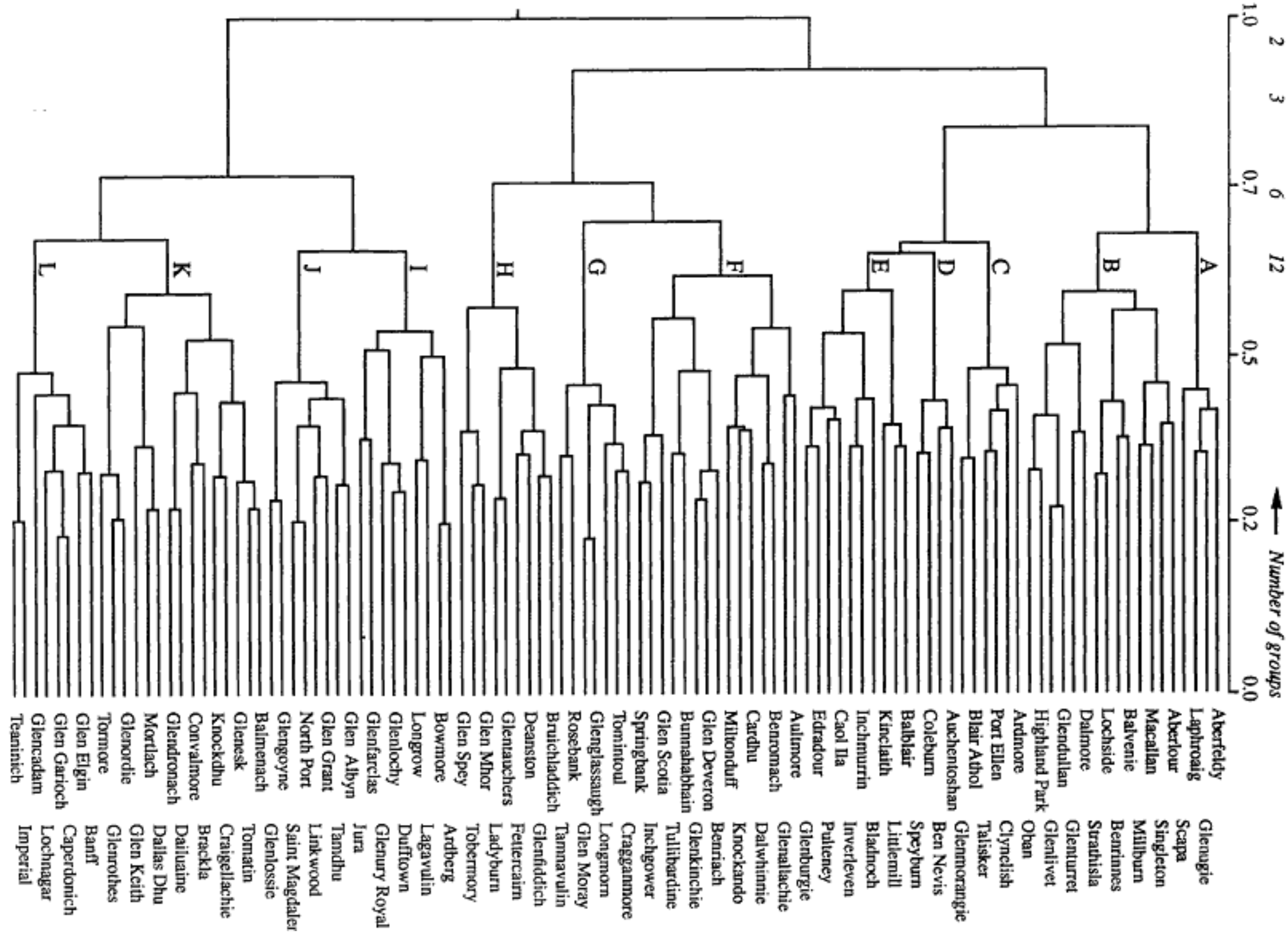
Figure 1. The Illicit Highland Whisky Still, by Sir Edwin Landseer, 1829. A family engaged in illicit distillation in a bothy in the hills, high above Glenlivet. Reproduced by courtesy of the Victoria and Albert Museum, London

David Wishart: The Flavour of Whisky, Significance, March 2009

# Robert Louis Stevenson:
# The Scotsman's Return From Abroad

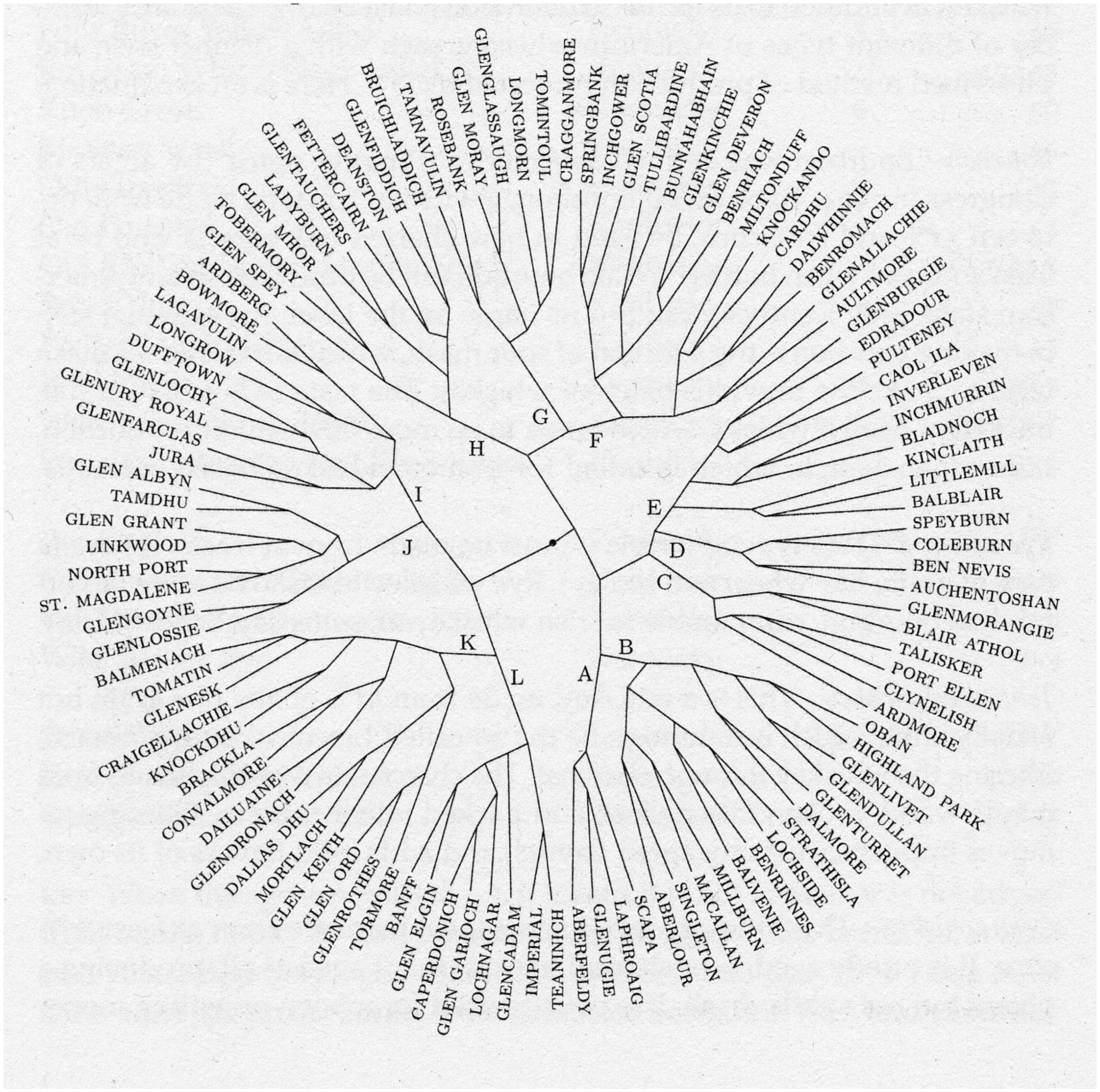
At last, across the weary faem,
Frae far, outlandish pairts I came.
On ilka side o' me I fand
Fresh tokens o' my native land.
Wi' whatna joy I hailed them a' –
The hilltaps standin' raw by raw,
The public house, the Hielan' birks,
And a' the bonny U.P. kirks!
But maistly thee, the bluid o' Scots,
Frae Maidenkirk to John o' Grots,
The king o' drinks, as I conceive it,
Talisker, Isla, or Glenlivet!

# Dendrogram of Scotch Single Malt Whiskies



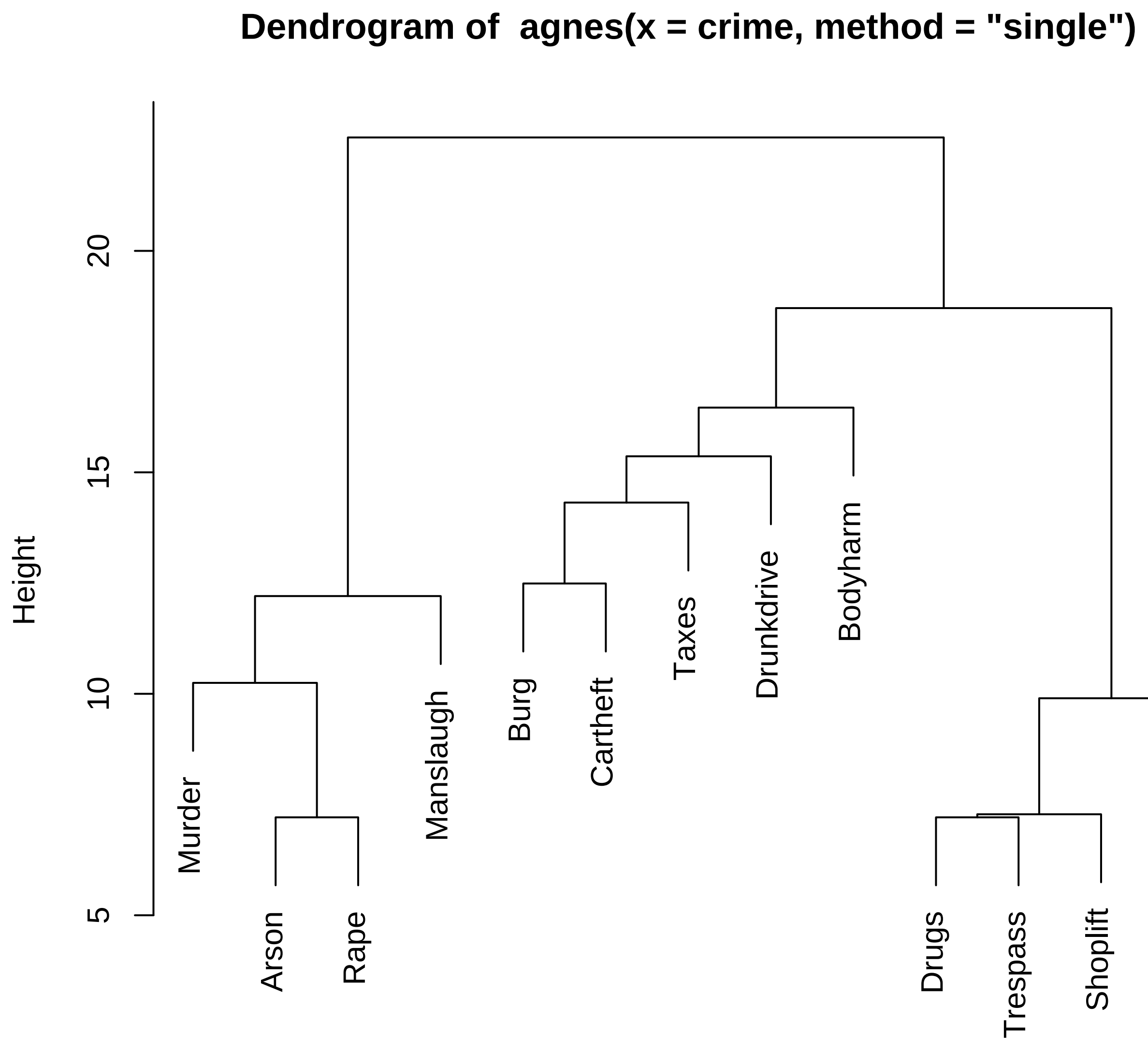
F. J. Lapointe & P. Legendre, Applied Statistics 43, 237-257 (1994)

# The Scotch revisited

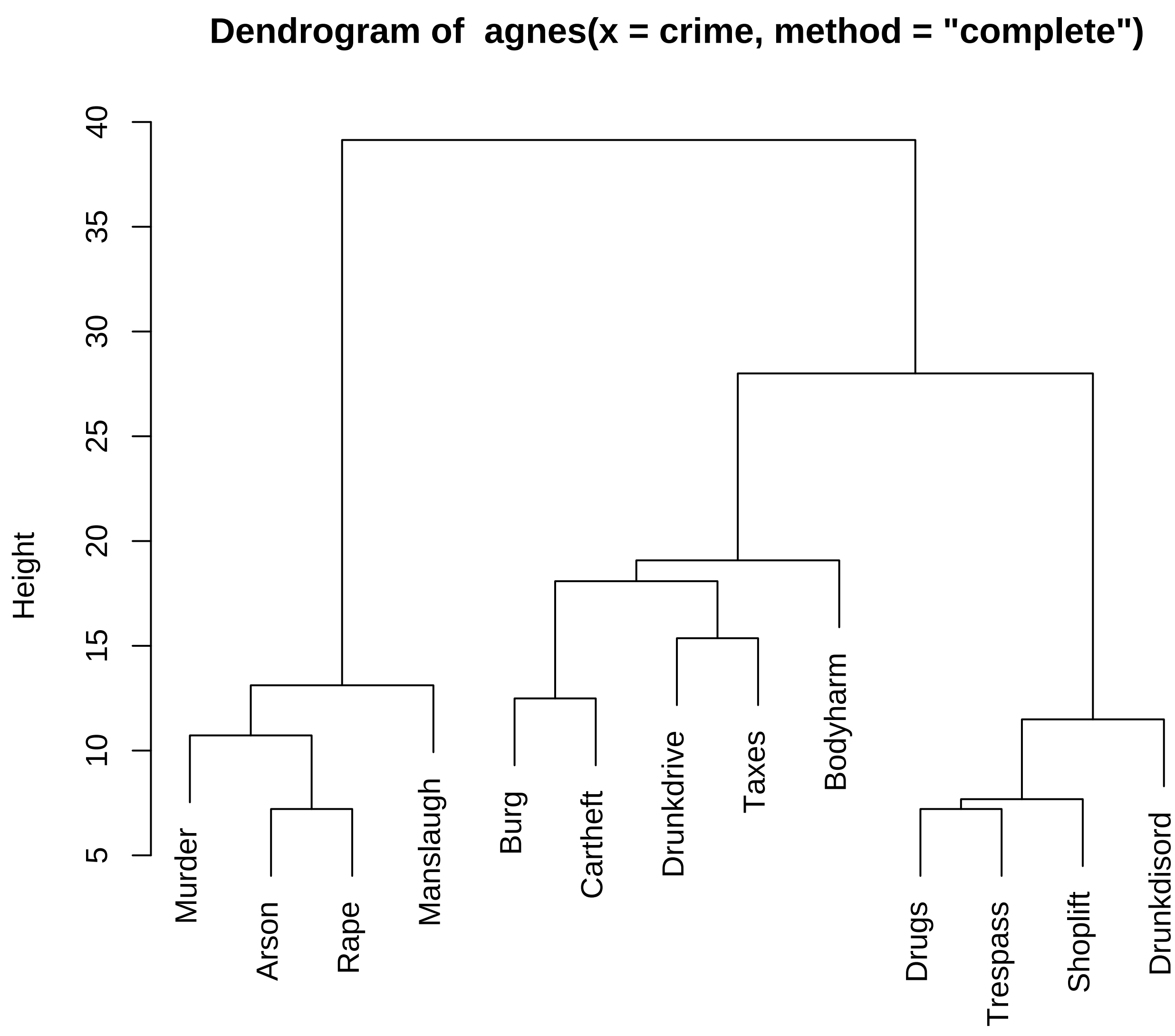


(Thomas Fink, The Man's Book, Little, Brown, and Company, New York, 2009 )

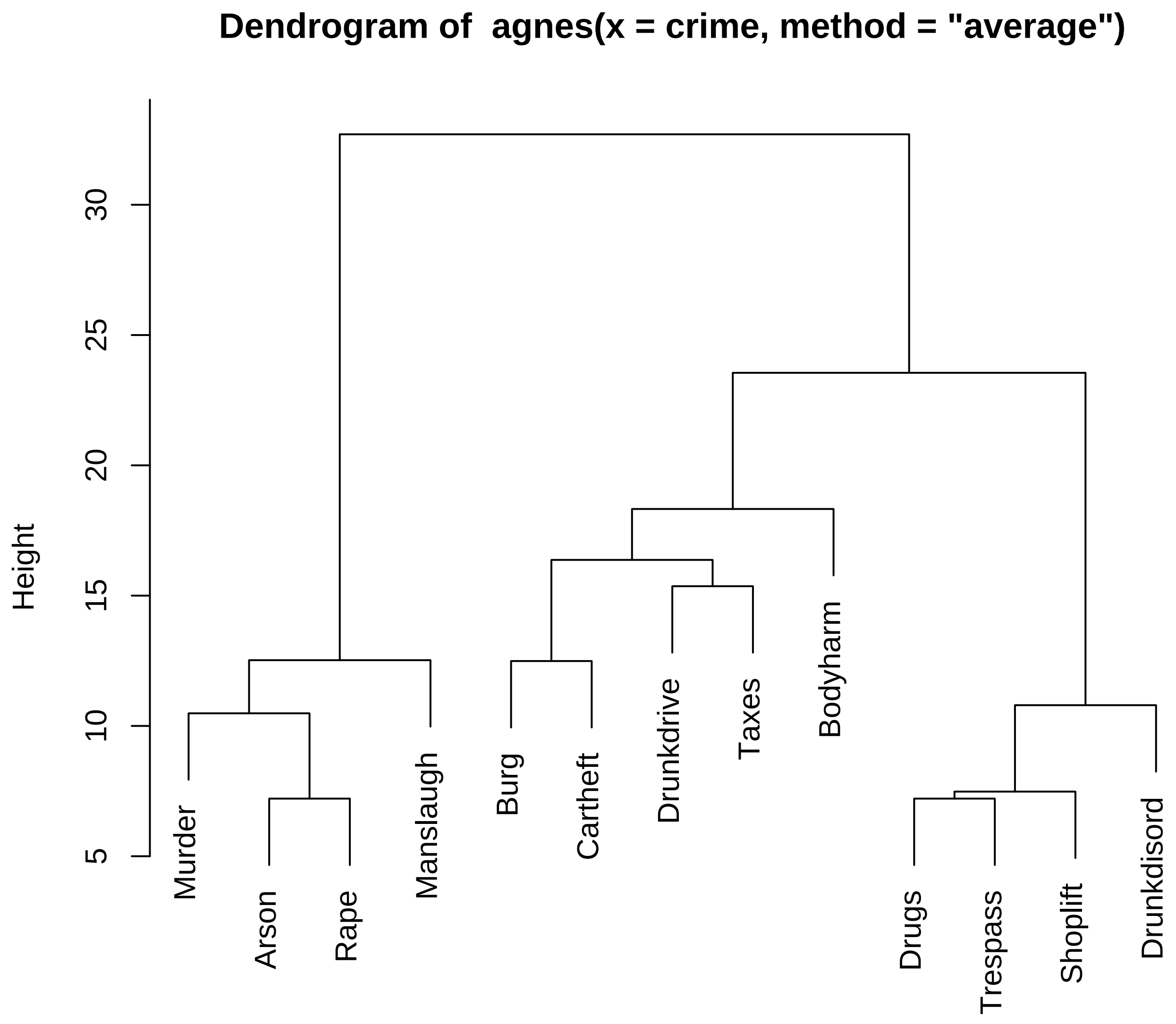
# Crimes, hierarchical: single



## Crimes, hierarchical: complete

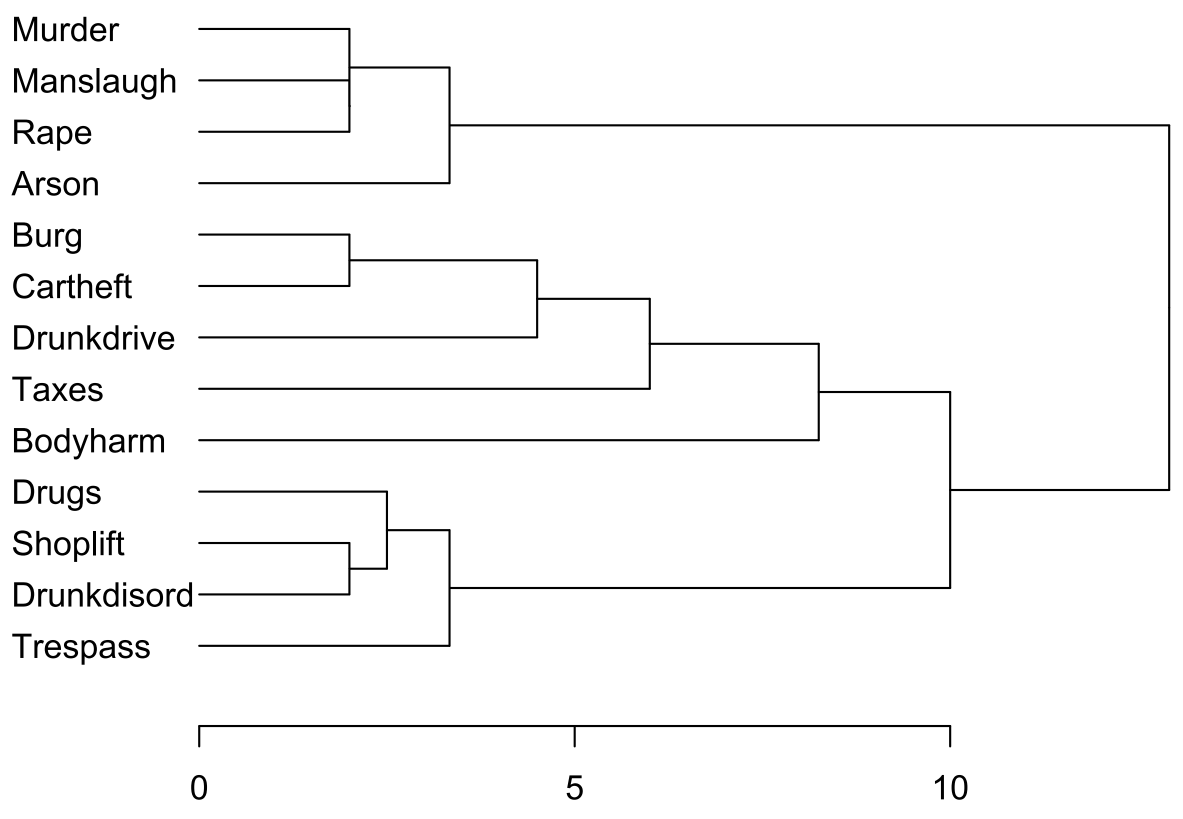


# Crimes, hierarchical: average



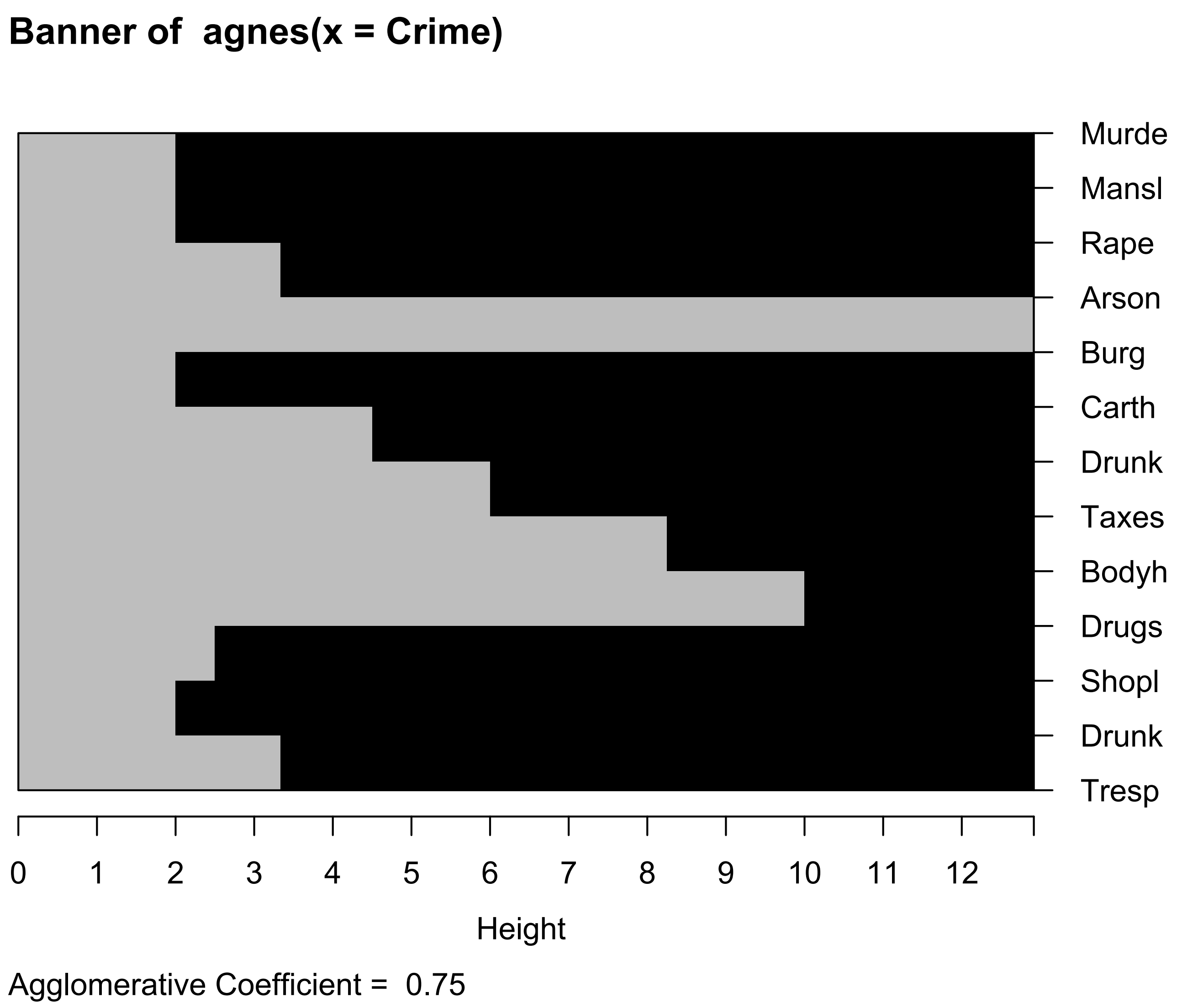
## Two styles of dendrogram plots: I

```
> plot(as.dendrogram(as.hclust(agnes(Crime))),horiz=T,
+ xlim=c(-2,13),ylim=c(14,0),dLeaf=-2.5)
```



# Two styles of dendrogram plots: II

```
> plot(agnes(Crime),which=1,col=c(8,1))
```



## Silhouette plot

For the observation with index $i$, define the silhouette width $s_i$

First, let $a_i$ - average dissimilarity between $i$-th object and all other points of the cluster to which $i$ belongs; if $i$-th object is the only one in its cluster, $s_i = 0$ without further calculations.

Then, for all other clusters $C$, let $d(i, C)$ be average dissimilarity of $i$-th object to all observations of $C$; the smallest of these $d(i, C)$ is $b_i = \min_C d(i, C)$, and can be seen as the dissimilarity between $i$-the object and its "neighbor" cluster, the nearest one to which it does not belong; and then

$$s_i = \frac{b_i - a_i}{\max\{a_i, b_i\}}$$

Observations with a large $s_i$ (almost 1) are very well clustered, a small $s_i$ (around 0) means that the observation lies between two clusters, and observations with a negative $s_i$ are probably placed in the wrong cluster.

Silhouette plot is primarily designed for partitioning methods, but may be used also for the hierarchical ones - which is facilitated by function `cutree`

## So

```
> plot(silhouette(pam(Crime,3)))
```

This would not work that nice:

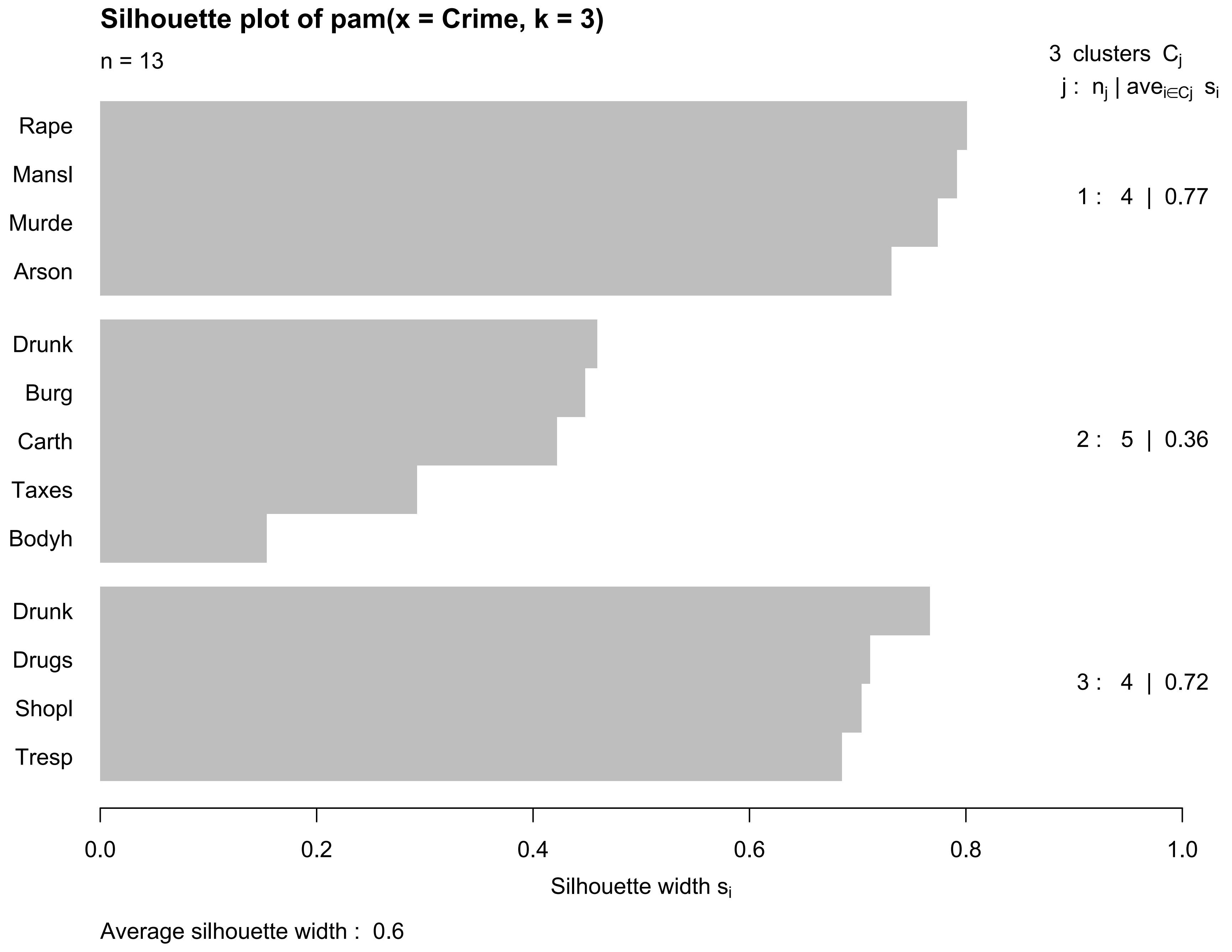
```
> plot(silhouette(cutree(agnes(Crime),3),dist=Crime))
```

But this makes it up:

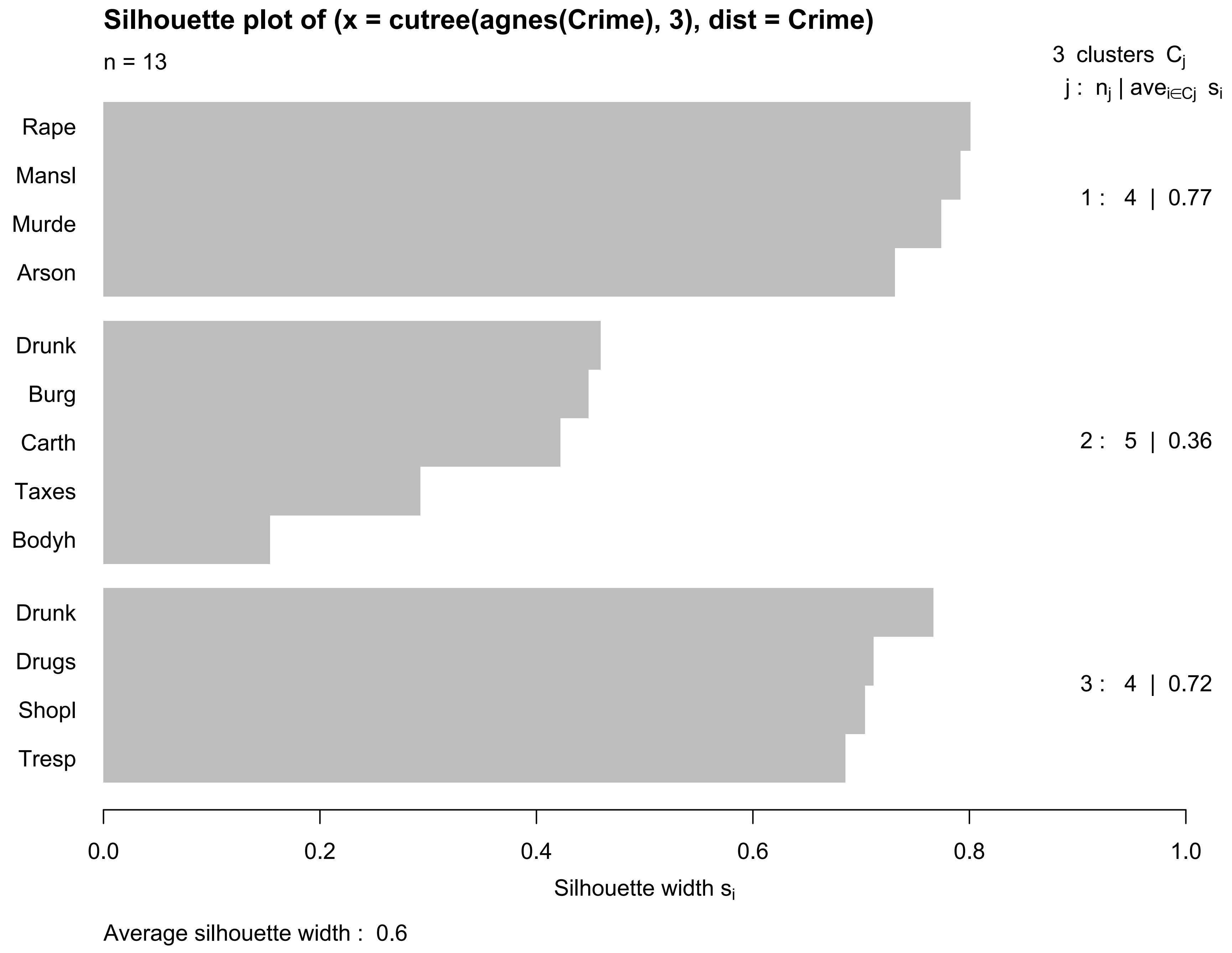
```
> sil <- silhouette(cutree(agnes(Crime),3),dist=Crime)
> row.names(sil) <- labels(Crime)
> plot(sil)
```

The silhouette plots come up as same - why?

# Silhouette for 3 medoids



# Silhouette for 3 agglomerative clusters (same)



## A possible twist of mind: clustering of variables

We can cluster variables in the same way as objects - once we can find a measure of dissimilarity for them

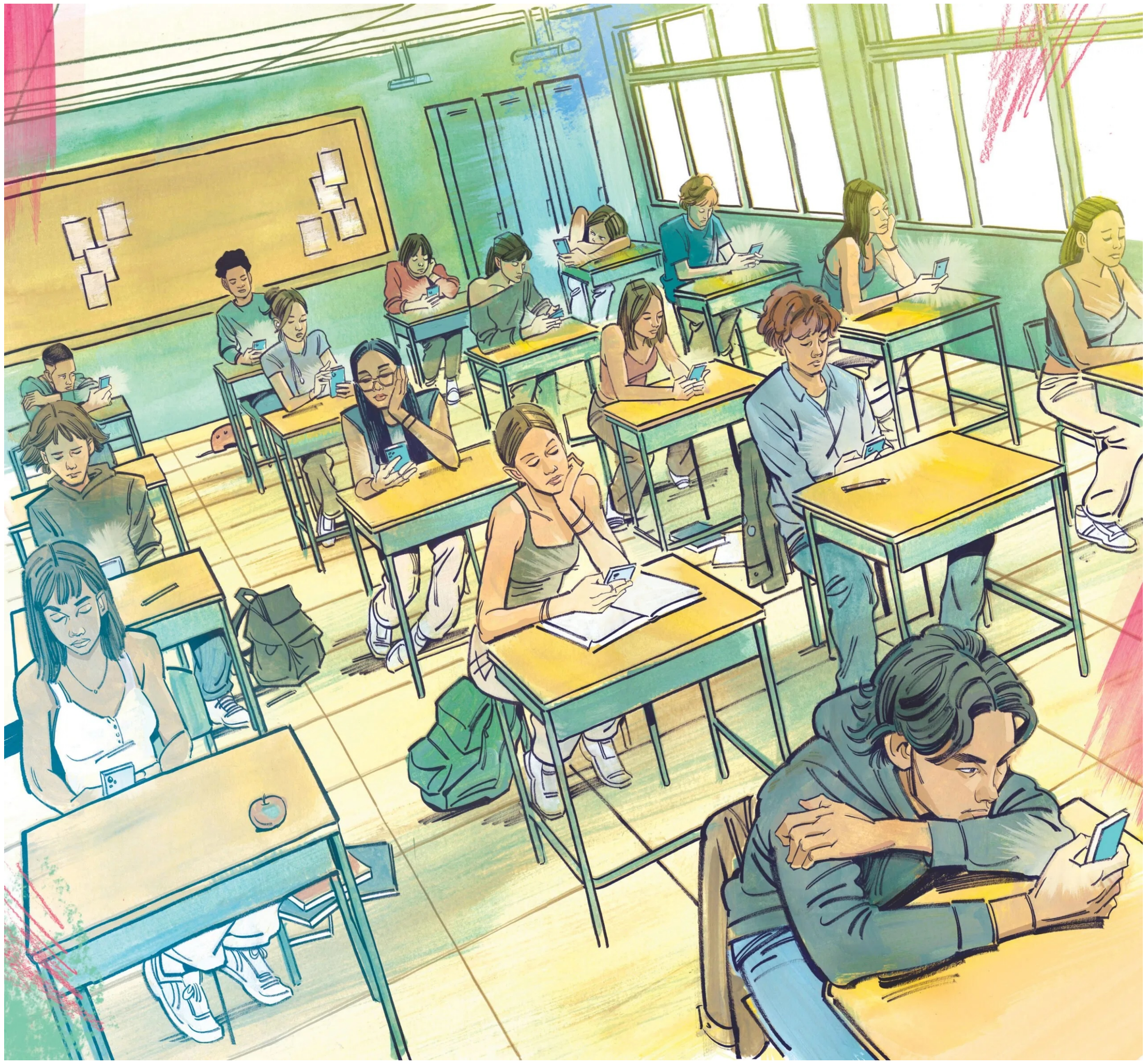
And the handy ones are those based on (various versions of) correlation coefficients

Of course, rather than $\rho$ itself it is $1 - \rho^2$ that qualifies as a dissimilarity

Or perhaps $1 - |\rho|$, or $1/(1 - \rho^2) - 1$, or $1/(1 - |\rho|) - 1$

But others may argue also for $1 - \rho$, or $1/(1 - \rho) - 1/2$

## Maclean’s 2004



# The data

```
           StudentBodyAverageEnteringGrade StudentProportionWith75PercOrHigher StudentProportionWhoGraduate
Toronto                               86.5                               99.78                         92.6
Queens                                89.0                              100.00                         91.6
McGill                                89.3                               99.77                         92.5
UBC                                   88.5                               99.55                         88.1
McMaster                              85.0                               99.98                         89.2
Dalhousie                             84.4                               96.85                         87.9
Western                               88.1                               99.96                         91.8
Alberta                               85.9                               96.31                         82.3
Ottawa                                83.7                               91.74                         90.3
Sask                                  86.5                               93.59                         86.8
Sherbrooke                            86.5                               97.89                         90.5
Montreal                              87.6                               99.30                         91.6
Laval                                 86.1                               97.02                         79.4
Manitoba                              83.1                               82.31                         84.8
Calgary                               82.6                               91.20                         78.6
           StudentOutOffProvince1stYear StudentInternationalGraduate StudentInternational1stYear StudentRetentionRate
Toronto                             3.8                         16.6                         6.2                 95.8
Queens                             14.2                         24.9                         2.2                 95.5
McGill                             30.0                         22.8                        17.9                 92.4
UBC                                13.4                         23.6                         8.6                 90.9
McMaster                            1.7                         21.0                         6.8                 90.1
Dalhousie                          52.6                         20.9                         6.1                 83.9
Western                             4.6                         14.5                         4.8                 94.9
Alberta                            15.2                         26.4                         3.4                 84.7
Ottawa                             13.5                         20.9                         5.8                 89.8
Sask                                5.8                         29.3                         0.8                 84.4
Sherbrooke                          3.6                         17.5                         1.4                 93.8
Montreal                            7.9                         20.8                        12.8                 90.0
Laval                               2.4                         20.3                        11.5                 95.2
Manitoba                            9.1                         17.4                         5.1                 85.1
Calgary                            12.9                         17.0                         1.9                 84.3
```

```
           StudentAwards ClassesClassSizes1stAnd2ndYearLevel Classes1stAnd2ndYearLevel1To25
Toronto              7.7                                3.18                           6.03
Queens               8.5                                3.34                           4.91
McGill               9.9                                3.54                           9.07
UBC                  7.8                                3.73                           9.82
McMaster             5.8                                3.19                           3.23
Dalhousie            6.9                                3.80                           7.53
Western              4.8                                4.06                          16.65
Alberta              7.1                                3.66                           7.75
Ottawa               4.6                                3.90                          10.36
Sask                 3.1                                4.19                          16.51
Sherbrooke           4.1                                4.74                          19.55
Montreal             6.0                                4.24                          14.26
Laval                5.7                                4.25                          20.82
Manitoba             5.2                                3.79                           7.40
Calgary              5.1                                3.70                           6.75
           Classes1stAnd2ndYearLevel26To50 Classes1stAnd2ndYearLevel51To100 Classes1stAnd2ndYearLevel101To250
Toronto                               9.37                            19.86                             35.25
Queens                                9.46                            24.51                             38.00
McGill                               13.43                            25.91                             32.72
UBC                                  15.62                            22.96                             42.31
McMaster                             11.29                            14.19                             44.43
Dalhousie                            19.71                            29.70                             33.15
Western                              19.72                            32.00                             18.69
Alberta                              15.61                            26.68                             34.43
Ottawa                               17.08                            29.24                             38.90
Sask                                 25.10                            25.58                             26.80
Sherbrooke                           41.59                            32.10                              6.76
Montreal                             19.70                            44.71                             18.45
Laval                                16.89                            31.25                             28.89
Manitoba                             17.54                            28.03                             41.02
Calgary                              15.22                            27.96                             41.48
           Classes1stAnd2ndYearLevel251To500 Classes1stAnd2ndYearLevelMore500 ClassesClassSizes3rdAnd4thYearLevel
Toronto                                20.23                             9.25                                4.55
Queens                                 21.80                             1.33                                4.60
McGill                                 11.34                             7.52                                4.93
```

```
UBC                                     8.21                             1.09                                 4.30
McMaster                               25.98                             0.88                                 4.29
Dalhousie                               8.32                             1.59                                 4.95
Western                                10.56                             2.38                                 5.08
Alberta                                15.54                             0.00                                 4.62
Ottawa                                  4.42                             0.00                                 4.76
Sask                                    6.01                             0.00                                 5.09
Sherbrooke                              0.00                             0.00                                 4.98
Montreal                                2.88                             0.00                                 4.83
Laval                                   2.16                             0.00                                 4.91
Manitoba                                5.99                             0.00                                 4.85
Calgary                                 8.60                             0.00                                 4.84
           Classes3rdAnd4thYearLevel1To25 Classes3rdAnd4thYearLevel26To50 Classes3rdAnd4thYearLevel51To100
Toronto                             22.25                           30.12                            30.74
Queens                              24.71                           30.94                            25.46
McGill                              33.07                           36.49                            23.04
UBC                                 15.14                           28.78                            32.29
McMaster                            19.87                           21.84                            28.87
Dalhousie                           34.92                           32.68                            25.11
Western                             39.69                           35.30                            17.96
Alberta                             26.32                           27.74                            28.43
Ottawa                              26.55                           31.36                            33.42
Sask                                35.97                           41.83                            17.67
Sherbrooke                          24.36                           50.92                            22.74
Montreal                            26.57                           35.90                            31.56
Laval                               36.51                           28.17                            24.76
Manitoba                            30.95                           29.27                            33.39
Calgary                             25.77                           40.73                            24.85
           Classes3rdAnd4thYearLevel101To250 Classes3rdAnd4thYearLevel251To500 Classes3rdAnd4thYearLevelMore500
Toronto                                14.53                              2.37                                0
Queens                                 17.54                              1.35                                0
McGill                                  5.65                              1.75                                0
UBC                                    18.37                              5.42                                0
McMaster                               26.11                              3.31                                0
Dalhousie                               7.29                              0.00                                0
Western                                 7.05                              0.00                                0
```

```
Alberta                             16.32                           1.19                               0
Ottawa                               8.66                           0.00                               0
Sask                                 4.53                           0.00                               0
Sherbrooke                           1.98                           0.00                               0
Montreal                             5.98                           0.00                               0
Laval                               10.56                           0.00                               0
Manitoba                             6.38                           0.00                               0
Calgary                              8.65                           0.00                               0
           ClassesTaughtByTenuredFaculty FacultyFacultyWithPhD AwardsPerFullTimeFaculty
Toronto                            68.90                  98.6                     10.8
Queens                             62.10                  95.3                     13.7
McGill                             49.50                  94.3                      8.9
UBC                                52.90                  98.9                      8.8
McMaster                           57.90                  96.9                      4.8
Dalhousie                          54.40                  92.7                      2.2
Western                            73.00                  98.3                      4.0
Alberta                            45.90                  98.9                      6.4
Ottawa                             58.70                  97.2                      5.6
Sask                               61.10                  93.7                      3.4
Sherbrooke                         66.80                  92.7                      3.5
Montreal                           60.10                  94.4                      5.9
Laval                              68.00                  97.8                      4.2
Manitoba                           44.30                  95.9                      1.8
Calgary                            61.39                  91.9                      3.6
           HumanitiesAverageGrantSizePerFaculty HumanitiesNumberOfGrantsPerFaculty
Toronto                                   20719                              57.17
Queens                                     8902                              29.79
McGill                                    19506                              61.96
UBC                                       15105                              47.99
McMaster                                   9706                              31.62
Dalhousie                                  7819                              27.90
Western                                    8311                              25.53
Alberta                                   10464                              32.80
Ottawa                                    11396                              33.48
Sask                                       7295                              18.40
Sherbrooke                                 5891                              14.50
```

```
Montreal                                           15629                           37.30
Laval                                               8531                           25.48
Manitoba                                            6289                           20.45
Calgary                                             5510                           19.89
           MedicalScienceGrantsAverageGrantSizePerFaculty MedicalScienceNumberOfGrantsPerFaculty
Toronto                                            126520                                 218.25
Queens                                              81948                                 103.21
McGill                                             111852                                 161.38
UBC                                                 92459                                 152.14
McMaster                                            78282                                 130.10
Dalhousie                                           43649                                  88.35
Western                                             84162                                 152.01
Alberta                                             84031                                 168.12
Ottawa                                              72827                                 126.93
Sask                                                37278                                  74.60
Sherbrooke                                          55142                                  99.42
Montreal                                            90428                                 133.57
Laval                                               74439                                 126.15
Manitoba                                            44526                                  82.80
Calgary                                             72167                                 118.49
           FinancesOperatingBudget FinancesScholarshipsBursariesPercOfBudget FinancesStudentServicesPercOfBudget
Toronto                       9878                                     13.29                                7.37
Queens                        8053                                     14.00                                4.60
McGill                        8860                                     11.34                                5.46
UBC                           7110                                     12.14                                8.05
McMaster                      8451                                     11.53                                6.53
Dalhousie                     8472                                     10.16                                4.04
Western                       8500                                     15.88                                5.16
Alberta                       9467                                     12.47                                4.77
Ottawa                        9532                                     10.53                                5.72
Sask                          9611                                      5.11                                3.66
Sherbrooke                    7836                                     11.81                                4.33
Montreal                      9490                                     10.20                                4.34
Laval                         8628                                      7.94                                4.42
Manitoba                      8108                                      5.42                                3.93
Calgary                       8684                                     10.26                                5.16
```

```
           LibraryTotalLibraryHoldingsThousand LibraryHoldingsPerStudent LibraryAcquisitions LibraryExpenses
Toronto                                  14468                       258               50.66            8.30
Queens                                    5402                       289               50.79            6.82
McGill                                    4919                       200               47.97            7.27
UBC                                       8903                       253               41.50            7.81
McMaster                                  3116                       156               46.35            5.25
Dalhousie                                 2252                       167               42.75            5.02
Western                                   7904                       283               58.10            7.74
Alberta                                  10029                       337               47.84            7.30
Ottawa                                    4279                       176               46.85            5.14
Sask                                      4387                       286               49.97            6.65
Sherbrooke                                1859                       138               49.54            4.63
Montreal                                  5202                       145               40.48            5.78
Laval                                     4716                       189               49.28            4.95
Manitoba                                  3211                       155               41.62            7.60
Calgary                                    509                       210               44.62            6.07
           ReputationAlumniSupport ReputationalRanking TotalRanking
Toronto                      26.80                99.0            1
Queens                       16.20                98.0            2
McGill                       18.80                97.9            3
UBC                          14.50                96.9            6
McMaster                     13.60                97.5            4
Dalhousie                    13.90                50.0           11
Western                      25.80                70.0            7
Alberta                      14.70                96.0            5
Ottawa                       13.50                35.0           14
Sask                         13.40                60.0            9
Sherbrooke                   18.50                60.3            8
Montreal                     15.90                55.0           10
Laval                        12.40                40.0           13
Manitoba                     14.80                30.0           15
Calgary                      14.09                45.0           12
```

# Maclean’s condensed

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Toronto | 86.5 | 99.78 | 92.6 | 3.8 | 16.6 | 6.2 | 95.8 | 7.7 | 3.18 | 6.03 | 9.37 | 19.86 | 35.25 | 20.23 | 9.25 | 4.55 | 22.25 | 30.12 | 30.74 |
| Queens | 89.0 | 100.00 | 91.6 | 14.2 | 24.9 | 2.2 | 95.5 | 8.5 | 3.34 | 4.91 | 9.46 | 24.51 | 38.00 | 21.80 | 1.33 | 4.60 | 24.71 | 30.94 | 25.46 |
| McGill | 89.3 | 99.77 | 92.5 | 30.0 | 22.8 | 17.9 | 92.4 | 9.9 | 3.54 | 9.07 | 13.43 | 25.91 | 32.72 | 11.34 | 7.52 | 4.93 | 33.07 | 36.49 | 23.04 |
| UBC | 88.5 | 99.55 | 88.1 | 13.4 | 23.6 | 8.6 | 90.9 | 7.8 | 3.73 | 9.82 | 15.62 | 22.96 | 42.31 | 8.21 | 1.09 | 4.30 | 15.14 | 28.78 | 32.29 |
| McMaster | 85.0 | 99.98 | 89.2 | 1.7 | 21.0 | 6.8 | 90.1 | 5.8 | 3.19 | 3.23 | 11.29 | 14.19 | 44.43 | 25.98 | 0.88 | 4.29 | 19.87 | 21.84 | 28.87 |
| Dalhousie | 84.4 | 96.85 | 87.9 | 52.6 | 20.9 | 6.1 | 83.9 | 6.9 | 3.80 | 7.53 | 19.71 | 29.70 | 33.15 | 8.32 | 1.59 | 4.95 | 34.92 | 32.68 | 25.11 |
| Western | 88.1 | 99.96 | 91.8 | 4.6 | 14.5 | 4.8 | 94.9 | 4.8 | 4.06 | 16.65 | 19.72 | 32.00 | 18.69 | 10.56 | 2.38 | 5.08 | 39.69 | 35.30 | 17.96 |
| Alberta | 85.9 | 96.31 | 82.3 | 15.2 | 26.4 | 3.4 | 84.7 | 7.1 | 3.66 | 7.75 | 15.61 | 26.68 | 34.43 | 15.54 | 0.00 | 4.62 | 26.32 | 27.74 | 28.43 |
| Ottawa | 83.7 | 91.74 | 90.3 | 13.5 | 20.9 | 5.8 | 89.8 | 4.6 | 3.90 | 10.36 | 17.08 | 29.24 | 38.90 | 4.42 | 0.00 | 4.76 | 26.55 | 31.36 | 33.42 |
| Sask | 86.5 | 93.59 | 86.8 | 5.8 | 29.3 | 0.8 | 84.4 | 3.1 | 4.19 | 16.51 | 25.10 | 25.58 | 26.80 | 6.01 | 0.00 | 5.09 | 35.97 | 41.83 | 17.67 |
| Sherbrooke | 86.5 | 97.89 | 90.5 | 3.6 | 17.5 | 1.4 | 93.8 | 4.1 | 4.74 | 19.55 | 41.59 | 32.10 | 6.76 | 0.00 | 0.00 | 4.98 | 24.36 | 50.92 | 22.74 |
| Montreal | 87.6 | 99.30 | 91.6 | 7.9 | 20.8 | 12.8 | 90.0 | 6.0 | 4.24 | 14.26 | 19.70 | 44.71 | 18.45 | 2.88 | 0.00 | 4.83 | 26.57 | 35.90 | 31.56 |
| Laval | 86.1 | 97.02 | 79.4 | 2.4 | 20.3 | 11.5 | 95.2 | 5.7 | 4.25 | 20.82 | 16.89 | 31.25 | 28.89 | 2.16 | 0.00 | 4.91 | 36.51 | 28.17 | 24.76 |
| Manitoba | 83.1 | 82.31 | 84.8 | 9.1 | 17.4 | 5.1 | 85.1 | 5.2 | 3.79 | 7.40 | 17.54 | 28.03 | 41.02 | 5.99 | 0.00 | 4.85 | 30.95 | 29.27 | 33.39 |
| Calgary | 82.6 | 91.20 | 78.6 | 12.9 | 17.0 | 1.9 | 84.3 | 5.1 | 3.70 | 6.75 | 15.22 | 27.96 | 41.48 | 8.60 | 0.00 | 4.84 | 25.77 | 40.73 | 24.85 |

| | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Toronto | 14.53 | 2.37 | 0 | 68.90 | 98.6 | 10.8 | 20719 | 57.17 | 126520 | 218.25 | 9878 | 13.29 | 7.37 | 14468 | 258 | 50.66 | 8.30 | 26.80 | 99.0 | 1 |
| Queens | 17.54 | 1.35 | 0 | 62.10 | 95.3 | 13.7 | 8902 | 29.79 | 81948 | 103.21 | 8053 | 14.00 | 4.60 | 5402 | 289 | 50.79 | 6.82 | 16.20 | 98.0 | 2 |
| McGill | 5.65 | 1.75 | 0 | 49.50 | 94.3 | 8.9 | 19506 | 61.96 | 111852 | 161.38 | 8860 | 11.34 | 5.46 | 4919 | 200 | 47.97 | 7.27 | 18.80 | 97.9 | 3 |
| UBC | 18.37 | 5.42 | 0 | 52.90 | 98.9 | 8.8 | 15105 | 47.99 | 92459 | 152.14 | 7110 | 12.14 | 8.05 | 8903 | 253 | 41.50 | 7.81 | 14.50 | 96.9 | 6 |
| McMaster | 26.11 | 3.31 | 0 | 57.90 | 96.9 | 4.8 | 9706 | 31.62 | 78282 | 130.10 | 8451 | 11.53 | 6.53 | 3116 | 156 | 46.35 | 5.25 | 13.60 | 97.5 | 4 |
| Dalhousie | 7.29 | 0.00 | 0 | 54.40 | 92.7 | 2.2 | 7819 | 27.90 | 43649 | 88.35 | 8472 | 10.16 | 4.04 | 2252 | 167 | 42.75 | 5.02 | 13.90 | 50.0 | 11 |
| Western | 7.05 | 0.00 | 0 | 73.00 | 98.3 | 4.0 | 8311 | 25.53 | 84162 | 152.01 | 8500 | 15.88 | 5.16 | 7904 | 283 | 58.10 | 7.74 | 25.80 | 70.0 | 7 |
| Alberta | 16.32 | 1.19 | 0 | 45.90 | 98.9 | 6.4 | 10464 | 32.80 | 84031 | 168.12 | 9467 | 12.47 | 4.77 | 10029 | 337 | 47.84 | 7.30 | 14.70 | 96.0 | 5 |
| Ottawa | 8.66 | 0.00 | 0 | 58.70 | 97.2 | 5.6 | 11396 | 33.48 | 72827 | 126.93 | 9532 | 10.53 | 5.72 | 4279 | 176 | 46.85 | 5.14 | 13.50 | 35.0 | 14 |
| Sask | 4.53 | 0.00 | 0 | 61.10 | 93.7 | 3.4 | 7295 | 18.40 | 37278 | 74.60 | 9611 | 5.11 | 3.66 | 4387 | 286 | 49.97 | 6.65 | 13.40 | 60.0 | 9 |
| Sherbrooke | 1.98 | 0.00 | 0 | 66.80 | 92.7 | 3.5 | 5891 | 14.50 | 55142 | 99.42 | 7836 | 11.81 | 4.33 | 1859 | 138 | 49.54 | 4.63 | 18.50 | 60.3 | 8 |
| Montreal | 5.98 | 0.00 | 0 | 60.10 | 94.4 | 5.9 | 15629 | 37.30 | 90428 | 133.57 | 9490 | 10.20 | 4.34 | 5202 | 145 | 40.48 | 5.78 | 15.90 | 55.0 | 10 |
| Laval | 10.56 | 0.00 | 0 | 68.00 | 97.8 | 4.2 | 8531 | 25.48 | 74439 | 126.15 | 8628 | 7.94 | 4.42 | 4716 | 189 | 49.28 | 4.95 | 12.40 | 40.0 | 13 |
| Manitoba | 6.38 | 0.00 | 0 | 44.30 | 95.9 | 1.8 | 6289 | 20.45 | 44526 | 82.80 | 8108 | 5.42 | 3.93 | 3211 | 155 | 41.62 | 7.60 | 14.80 | 30.0 | 15 |
| Calgary | 8.65 | 0.00 | 0 | 61.39 | 91.9 | 3.6 | 5510 | 19.89 | 72167 | 118.49 | 8684 | 10.26 | 5.16 | 509 | 210 | 44.62 | 6.07 | 14.09 | 45.0 | 12 |

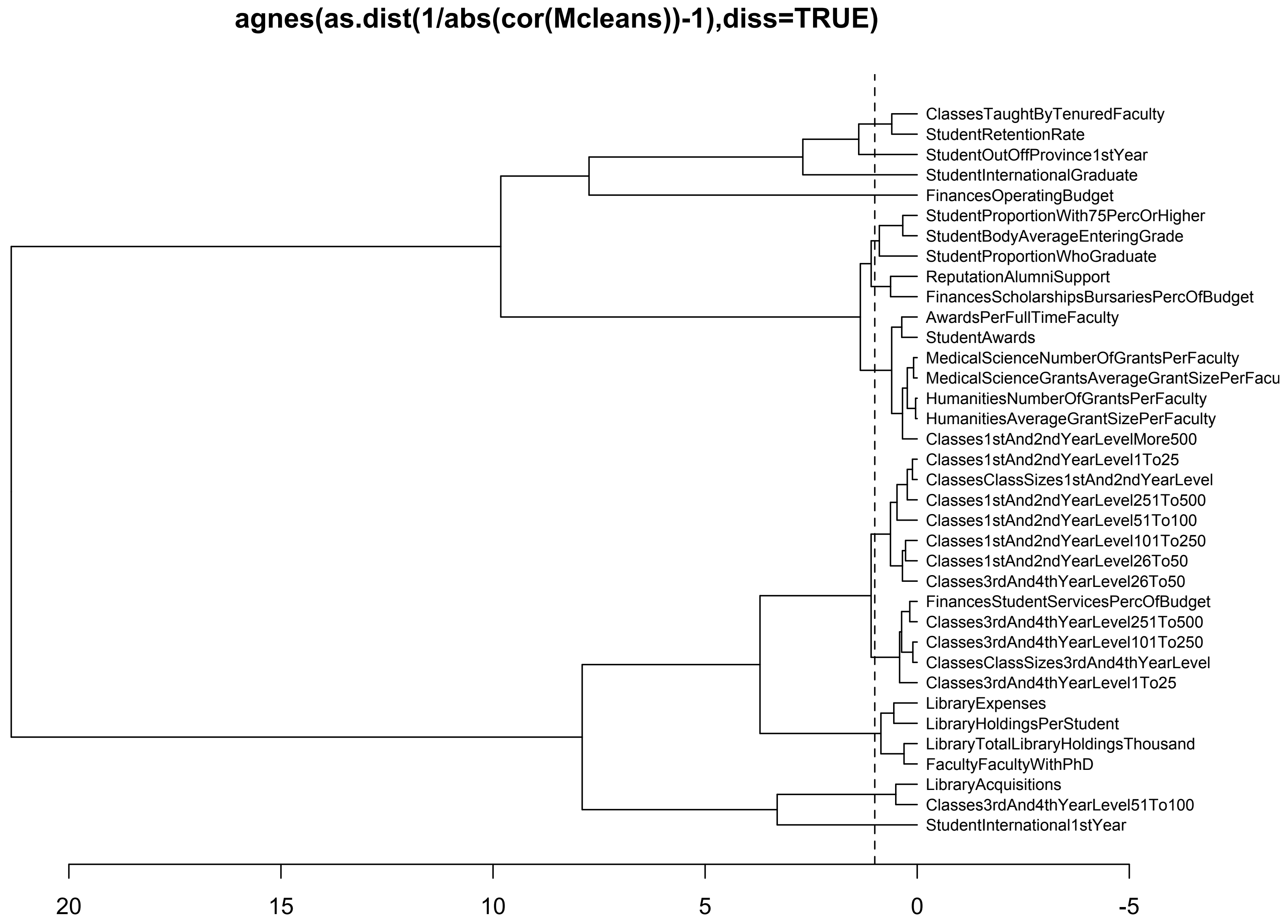
## Universities and variables

```
> row.names(Macleans)
 [1]  "Toronto"    "Queens"     "McGill"     "UBC"        "McMaster"
 [6]  "Dalhousie"  "Western"    "Alberta"    "Ottawa"     "Sask"
 [11] "Sherbrooke" "Montreal"   "Laval"      "Manitoba"   "Calgary"
> names(Macleans)
 [1] "StudentBodyAverageEnteringGrade"           "StudentProportionWith75PercOrHigher"
 [3] "StudentProportionWhoGraduate"              "StudentOutOffProvince1stYear"
 [5] "StudentInternationalGraduate"              "StudentInternational1stYear"
 [7] "StudentRetentionRate"                      "StudentAwards"
 [9] "ClassesClassSizes1stAnd2ndYearLevel"       "Classes1stAnd2ndYearLevel1To25"
[11] "Classes1stAnd2ndYearLevel26To50"           "Classes1stAnd2ndYearLevel51To100"
[13] "Classes1stAnd2ndYearLevel101To250"         "Classes1stAnd2ndYearLevel251To500"
[15] "Classes1stAnd2ndYearLevelMore500"          "ClassesClassSizes3rdAnd4thYearLevel"
[17] "Classes3rdAnd4thYearLevel1To25"            "Classes3rdAnd4thYearLevel26To50"
[19] "Classes3rdAnd4thYearLevel51To100"          "Classes3rdAnd4thYearLevel101To250"
[21] "Classes3rdAnd4thYearLevel251To500"         "Classes3rdAnd4thYearLevelMore500"
[23] "ClassesTaughtByTenuredFaculty"             "FacultyFacultyWithPhD"
[25] "AwardsPerFullTimeFaculty"                  "HumanitiesAverageGrantSizePerFaculty"
[27] "HumanitiesNumberOfGrantsPerFaculty"        "MedicalScienceGrantsAverageGrantSizePerFaculty"
[29] "MedicalScienceNumberOfGrantsPerFaculty"    "FinancesOperatingBudget"
[31] "FinancesScholarshipsBursariesPercOfBudget" "FinancesStudentServicesPercOfBudget"
[33] "LibraryTotalLibraryHoldingsThousand"       "LibraryHoldingsPerStudent"
[35] "LibraryAcquisitions"                       "LibraryExpenses"
[37] "ReputationAlumniSupport"                   "ReputationalRanking"
[39] "TotalRanking"
```

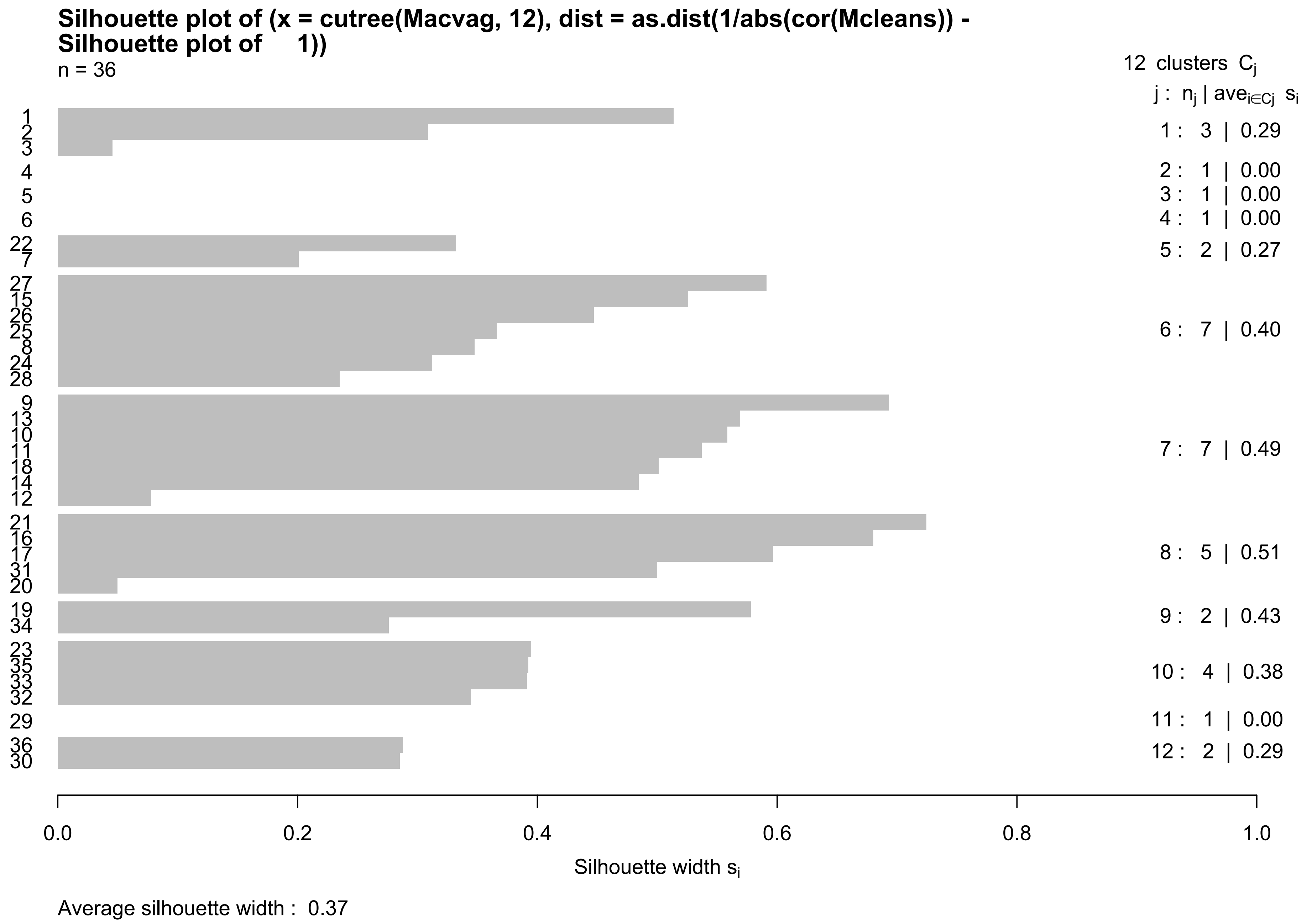
## Variables omitted from the analysis

```
> Macleans[order(Macleans[,38],decreasing=TRUE),c(22,38,39)]

           Classes3rdAnd4thYearLevelMore500 ReputationalRanking TotalRanking
Toronto                                   0                99.0            1
Queens                                    0                98.0            2
McGill                                    0                97.9            3
McMaster                                  0                97.5            4
UBC                                       0                96.9            6
Alberta                                   0                96.0            5
Western                                   0                70.0            7
Sherbrooke                                0                60.3            8
Sask                                      0                60.0            9
Montreal                                  0                55.0           10
Dalhousie                                 0                50.0           11
Calgary                                   0                45.0           12
Laval                                     0                40.0           13
Ottawa                                    0                35.0           14
Manitoba                                  0                30.0           15

> Mcleans=Macleans[,-c(22,38,39)]
```

# Hierarchical clustering with $1/|\rho| - 1$



# 12 clusters, perhaps



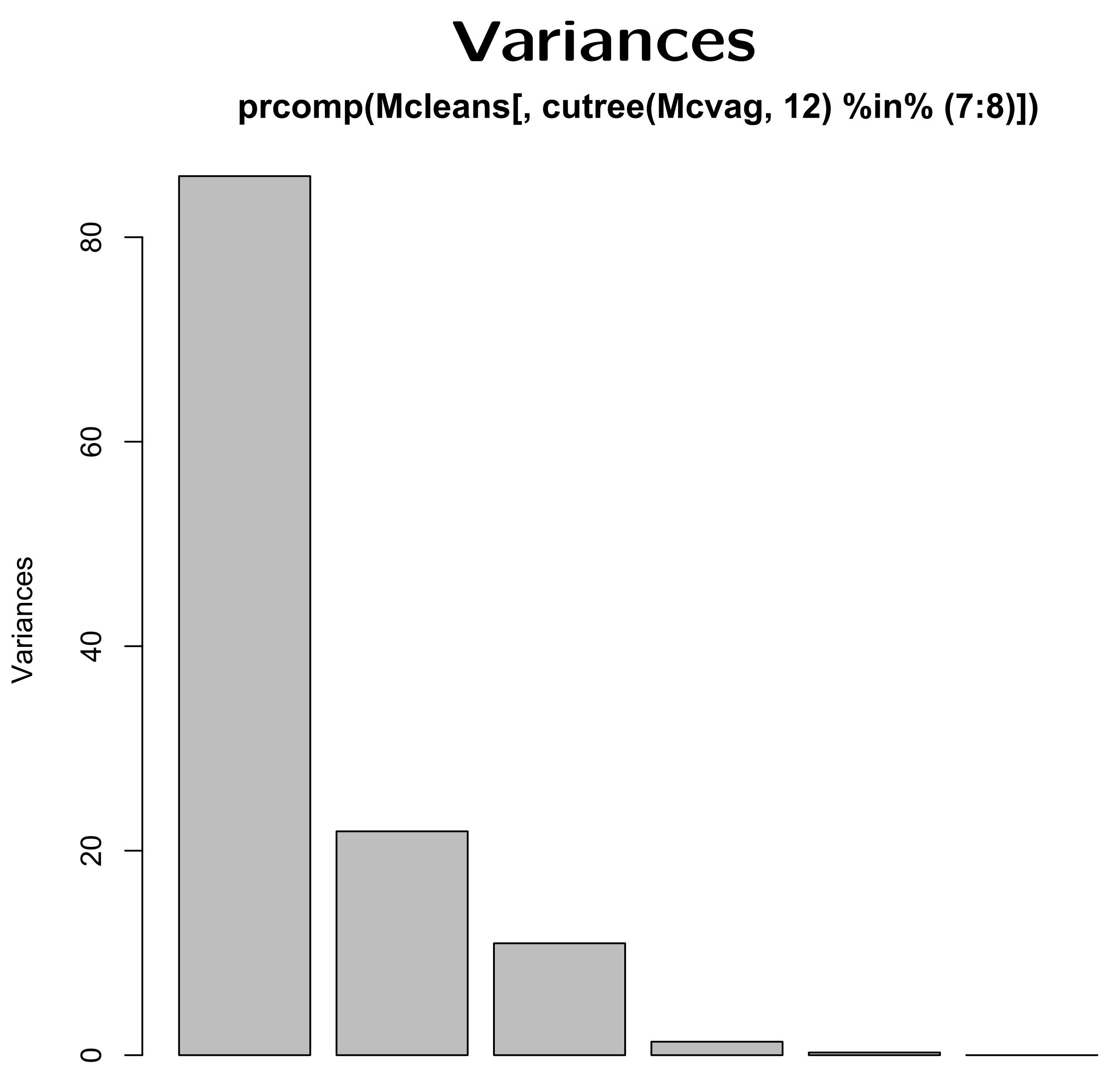
## The clusters

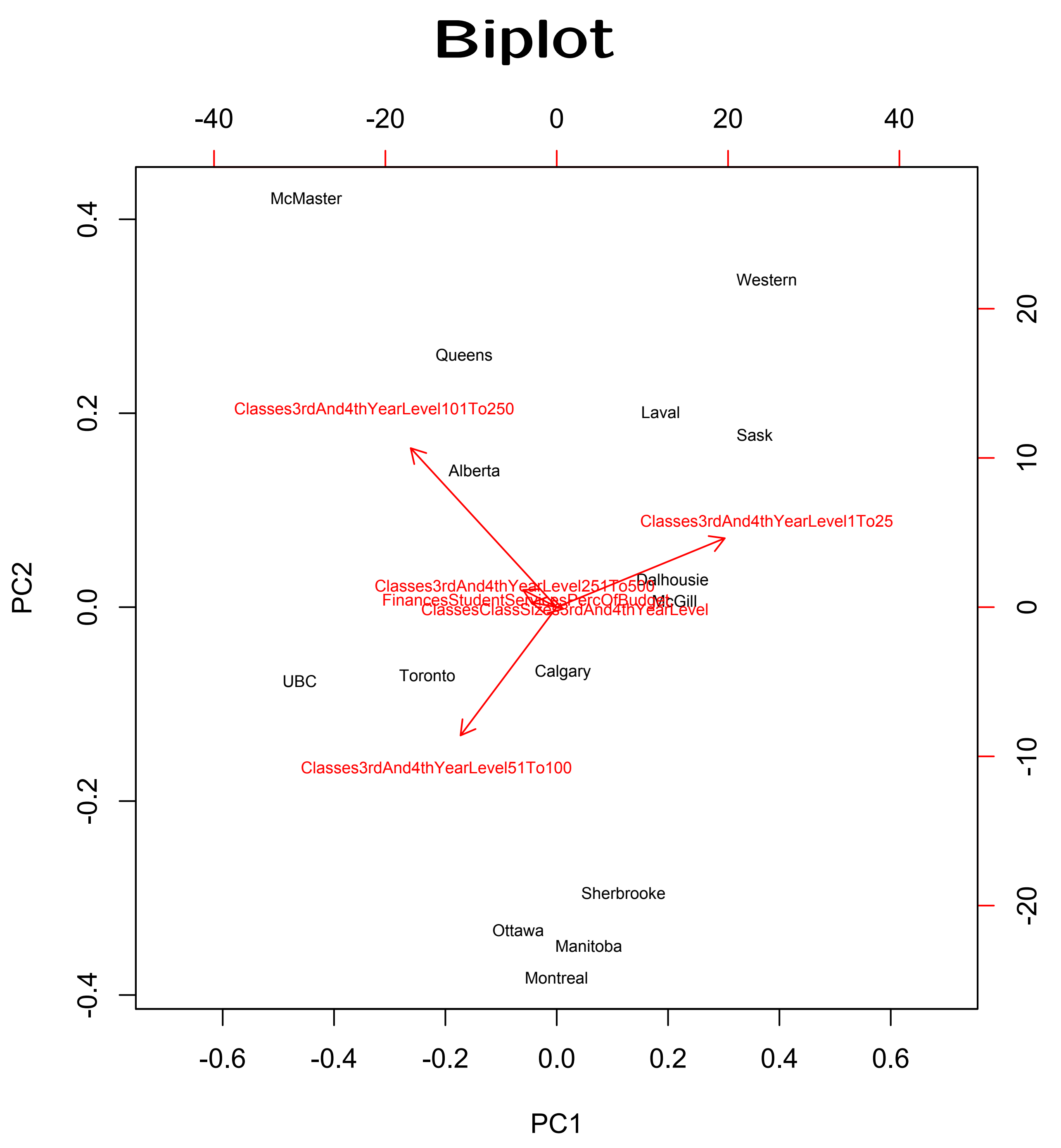
```
1| "StudentBodyAverageEnteringGrade"     "StudentProportionWith75PercOrHigher"
   "StudentProportionWhoGraduate"
2| "StudentOutOffProvince1stYear"
3| "StudentInternationalGraduate"
4| "StudentInternational1stYear"
5| "StudentRetentionRate"                "ClassesTaughtByTenuredFaculty"
6| "StudentAwards"                       "Classes1stAnd2ndYearLevelMore500"
   "AwardsPerFullTimeFaculty"            "HumanitiesAverageGrantSizePerFaculty"
   "HumanitiesNumberOfGrantsPerFaculty"  "MedicalScienceGrantsAverageGrantSizePerF
   "MedicalScienceNumberOfGrantsPerFaculty"
7| "ClassesClassSizes1stAnd2ndYearLevel" "Classes1stAnd2ndYearLevel1To25"
   "Classes1stAnd2ndYearLevel26To50"     "Classes1stAnd2ndYearLevel51To100"
   "Classes1stAnd2ndYearLevel101To250"   "Classes1stAnd2ndYearLevel251To500"
   "Classes3rdAnd4thYearLevel26To50"
8| "ClassesClassSizes3rdAnd4thYearLevel" "Classes3rdAnd4thYearLevel1To25"
   "Classes3rdAnd4thYearLevel101To250"   "Classes3rdAnd4thYearLevel251To500"
   "FinancesStudentServicesPercOfBudget"
9| "Classes3rdAnd4thYearLevel51To100"    "LibraryAcquisitions"
10|"FacultyFacultyWithPhD"               "LibraryTotalLibraryHoldingsThousand"
   "LibraryHoldingsPerStudent"           "LibraryExpenses"
11|"FinancesOperatingBudget"
12|"FinancesScholarshipsBursariesPercOfBudget" "ReputationAlumniSupport"
```

## Class size variables

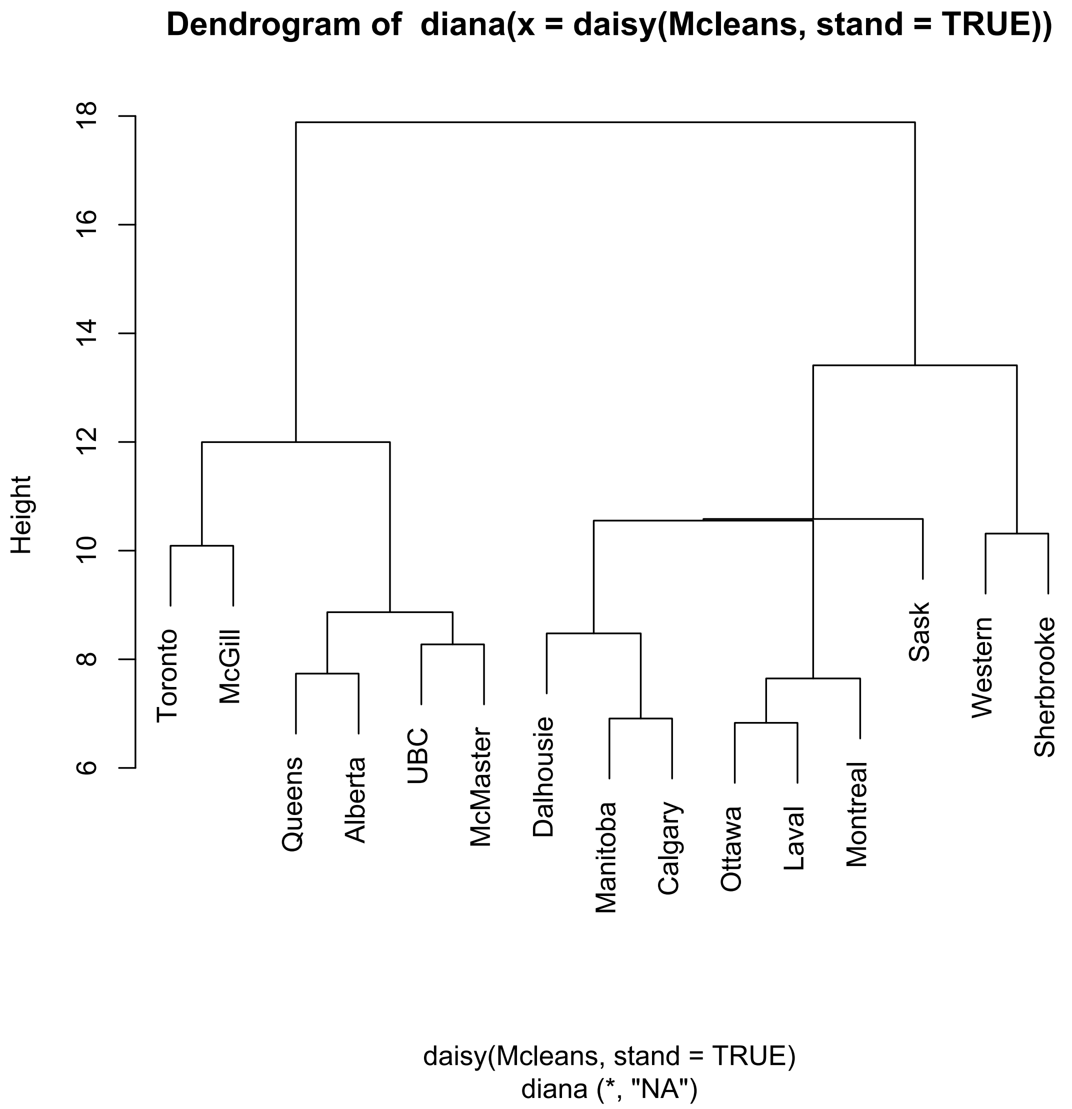
```
 [1] "ClassesClassSizes1stAnd2ndYearLevel" "Classes1stAnd2ndYearLevel1To25"
 [3] "Classes1stAnd2ndYearLevel26To50"     "Classes1stAnd2ndYearLevel51To100"
 [5] "Classes1stAnd2ndYearLevel101To250"   "Classes1stAnd2ndYearLevel251To500"
 [7] "ClassesClassSizes3rdAnd4thYearLevel" "Classes3rdAnd4thYearLevel1To25"
 [9] "Classes3rdAnd4thYearLevel26To50"     "Classes3rdAnd4thYearLevel101To250"
[11] "Classes3rdAnd4thYearLevel251To500"   "FinancesStudentServicesPercOfBudget"
```

```
> plot(prcomp(Mcleans[,cutree(Mcvag,12) %in% (7:8)]))
> biplot(prcomp(Mcleans[,cutree(Mcvag,12) %in% (7:8)]),
+ cex=0.6,xlim=c(-.7,.7))
```


Variances
prcomp(Mcleans[, cutree(Mcvag, 12) %in% (7:8)])
Variances
0
20
40
60
80


Biplot
PC1
PC2
McMaster
Western
Queens
Laval
Sask
Alberta
Dalhousie
McGill
UBC
Toronto
Calgary
Sherbrooke
Ottawa
Manitoba
Montreal
Classes3rdAnd4thYearLevel101To250
Classes3rdAnd4thYearLevel1To25
Classes3rdAnd4thYearLevel251To500
FinancesStudentServicesPercOfBudget
ClassesClassSize3rdAnd4thYearLevel
Classes3rdAnd4thYearLevel51To100

## How about universities?

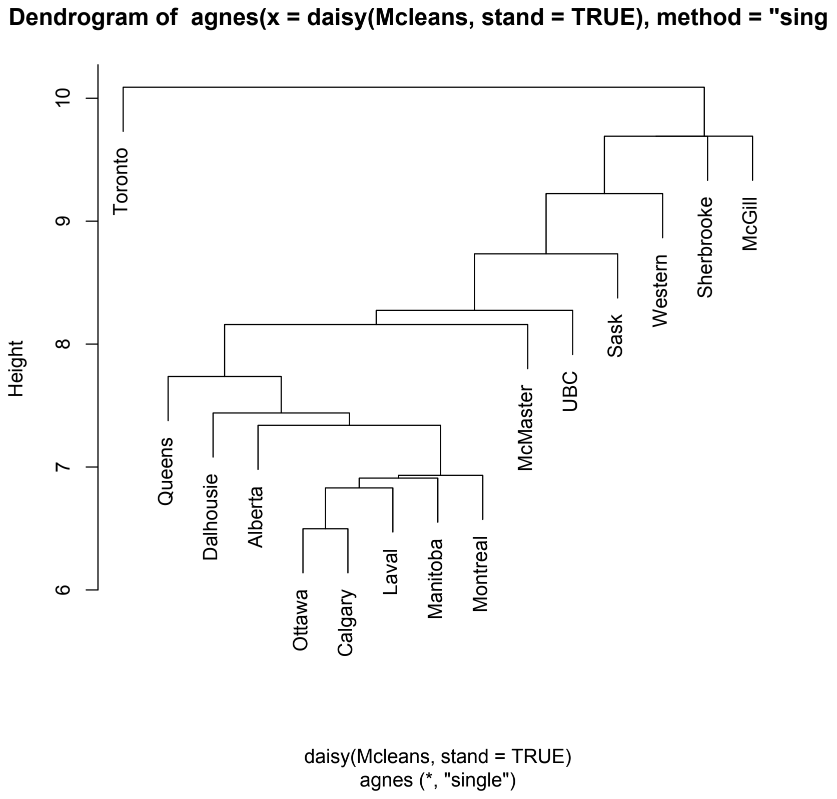


Divisive hierarchical clustering - the other versions yield similar results, except for the single linkage - see next page, also, for the explanation for the “doubled lines” in the right-hand side

# Odds and ends

1.
```
> diana(daisy(Mcleans,stand=TRUE))$height
 [1] 10.089735  11.997787   7.736485   8.867114  8.274415
 [61  7.885711   8.476898   6.909834  10.552828  6.830103
 [11] 7.647387  10.582905  13.411259  10.313927
```

2.



3. Note: standardization is a must here, otherwise we would cluster pretty much in the original variables 26, 28, 30, and 33.

## Model-based clustering

Distance-base methods - work well when distances can be trusted...

For instance, Euclidean distance can substantially depend on the scaling of the original variables

Fitting mixtures of distributions (multivariate normal)