

Aims and relevant data

We can make a dichotomy of data-mining and multivariate statistical methods into two groups, one series of methods confers a particular status to one variable or set of variables, these are to be predicted or explained, they are the response. Methods include regression, multiple response regression, discriminant analysis, analysis of variance depending on whether the explanatory variables are categorical or continuous. These are not the ones we are going to study here.

Table of Methods for Studying Links between Variables

Techniques	Vars. to explain (response)	Explanatory Var.
Multiple Regression	1 continuous	p continuous
Analysis of Variance	1 continuous	p categorical
Analysis of Covariance	1 continuous	p1 continuous, p2 categorical
Correspondence Analysis	1 categorical	1 categorical
Canonical Correl. Analysis	q continuous	p continuous
Principal Components w/r/t Instrumental Variables	q continuous	p continuous
Discriminant Analysis	1 categorical	p continuous
Multidimensional Analysis of Variance	p continuous	p categorical
Multidimensional Analysis of Covariance	p continuous	p1 categorical p2 continuous
Regression Tree	1 continuous	p1 continuous p2 categorical
Classification Tree	1 categorical	p1 continuous p2 categorical

Table of Methods for Representing Data: Techniques

Variables

Principal Components

p continuous

Multiple Correspondence Analysis

p categorical or

p categorical and q continuous

Multidimensionnal Scaling

categoricals and continuous

Clustering (either hierarchical or not)

continuous and categorical

Other dichotomies are possible:

- Bayesian/Frequentists.
- Parametric/ Nonparametric.
- Robust.
- Supervised/ Unsupervised.
- Exploratory/ Confirmatory.

Fisher's Classical Paradigm This will be explained as a flow chart showing where Fisher's initial hypotheses come in and how the data are studied under the conjectured distribution : usually multivariate normal. There are several books on Multivariate Analysis that follow this paradigm and I will not go into any details, but it makes sense to bear in mind that if one does know from previous studies for instance that the data will be normal multivariate, there is only a simple estimation problem left, all the possible information from the data is summarised by the mean and variance covariance matrix and no more is to be said.

Tukey and Mallow's Exploratory-Confirmatory Paradigm

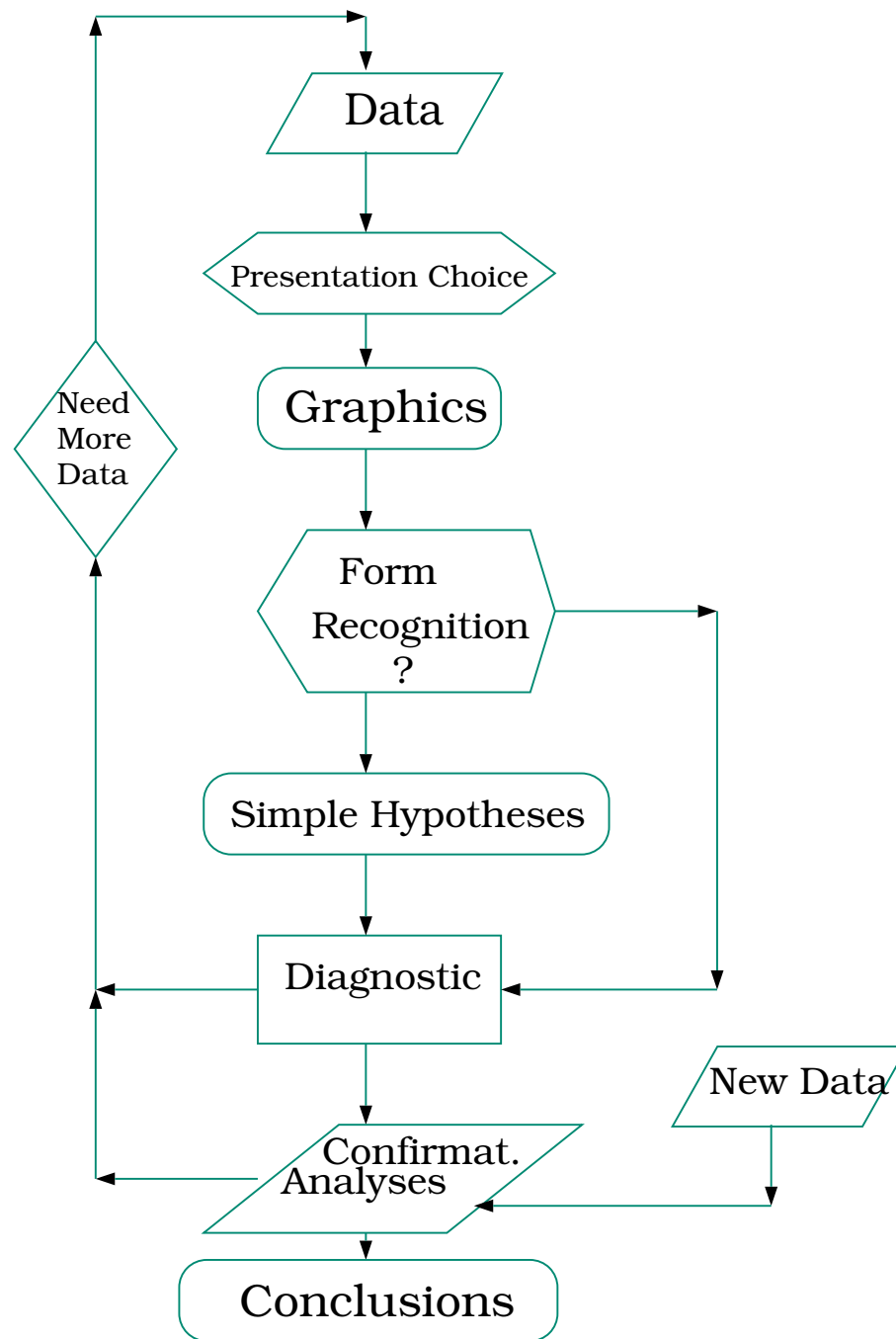
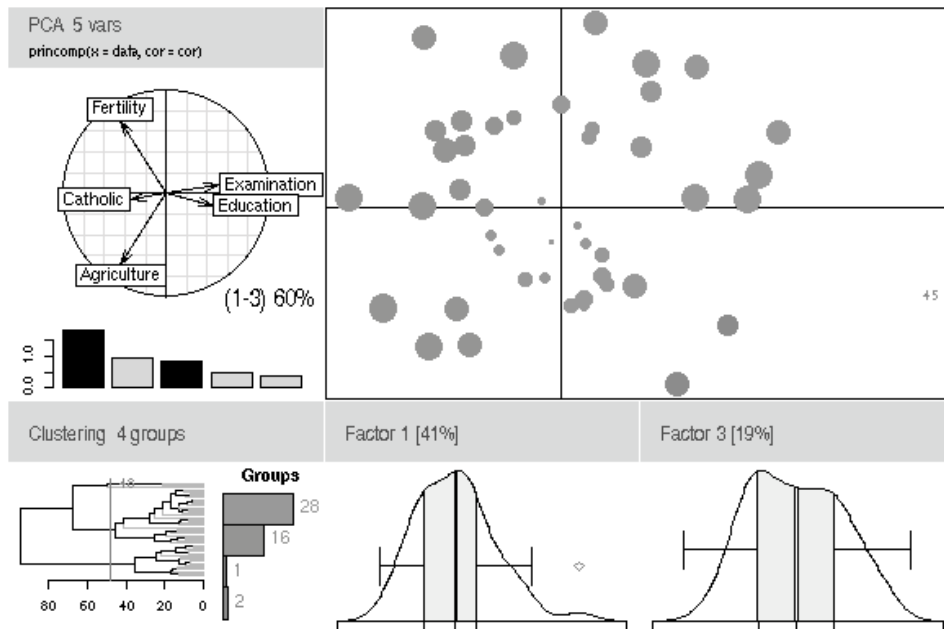


Figure 1: Modern Paradigm

Visualization Tools



The best software for visualizing data is provided by R an opensource package that has a large Exploratory Data Analysis component provided by the AT&T labs under the guidance of the master of data visualization: John Tukey who invented the term EDA, stem and leaf plots, boxplots, projection pursuit (with here present Jerry Friedman), and many more brilliant visualization wonders.

Distribution evaluation

Histograms are useless, they do not provide us with a good visual evaluation of a distribution pictures instead of one number summaries.

- Histograms.
- QQplots.
- PPlots.

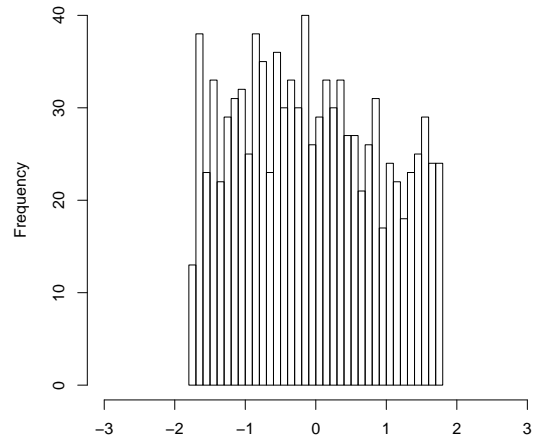
Random Matrix Data

QR decomposition, Gram-Schmidt decomposition of iid uniform data.

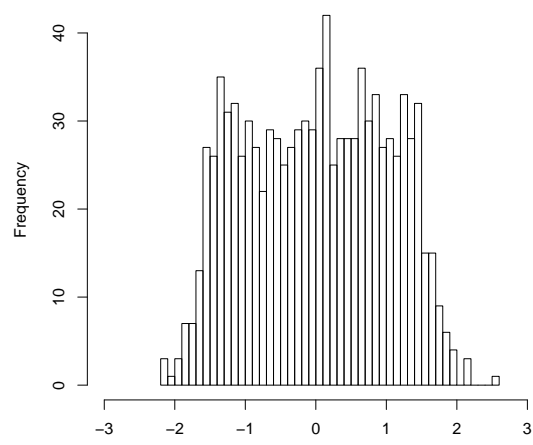
We fill a 1000×1000 data matrix according to a $\text{Uniform}(-1,1)$.

We then find the QR decomposition of the matrix and multiply Q by $\sqrt{1000}$, as we follow the columns the data become more and more normal.

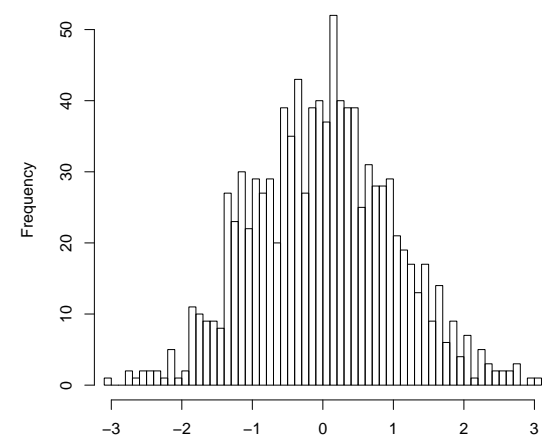
QR Column 1



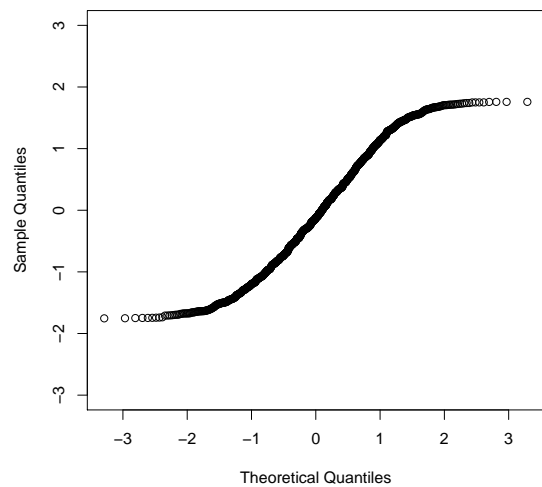
QR Column 101



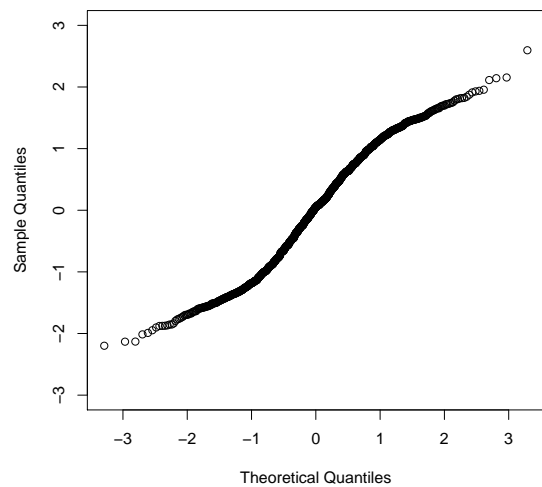
QR Column 501



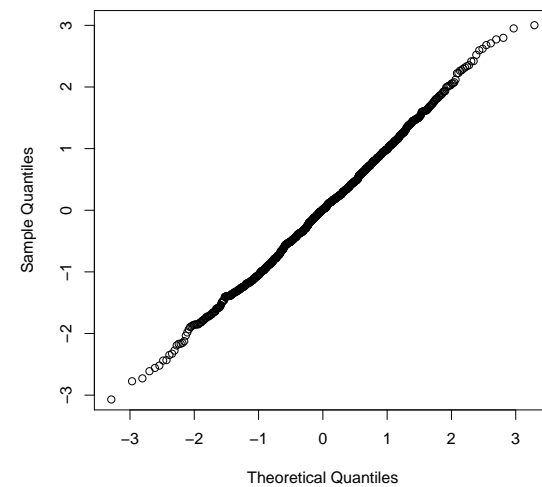
QR Column 1



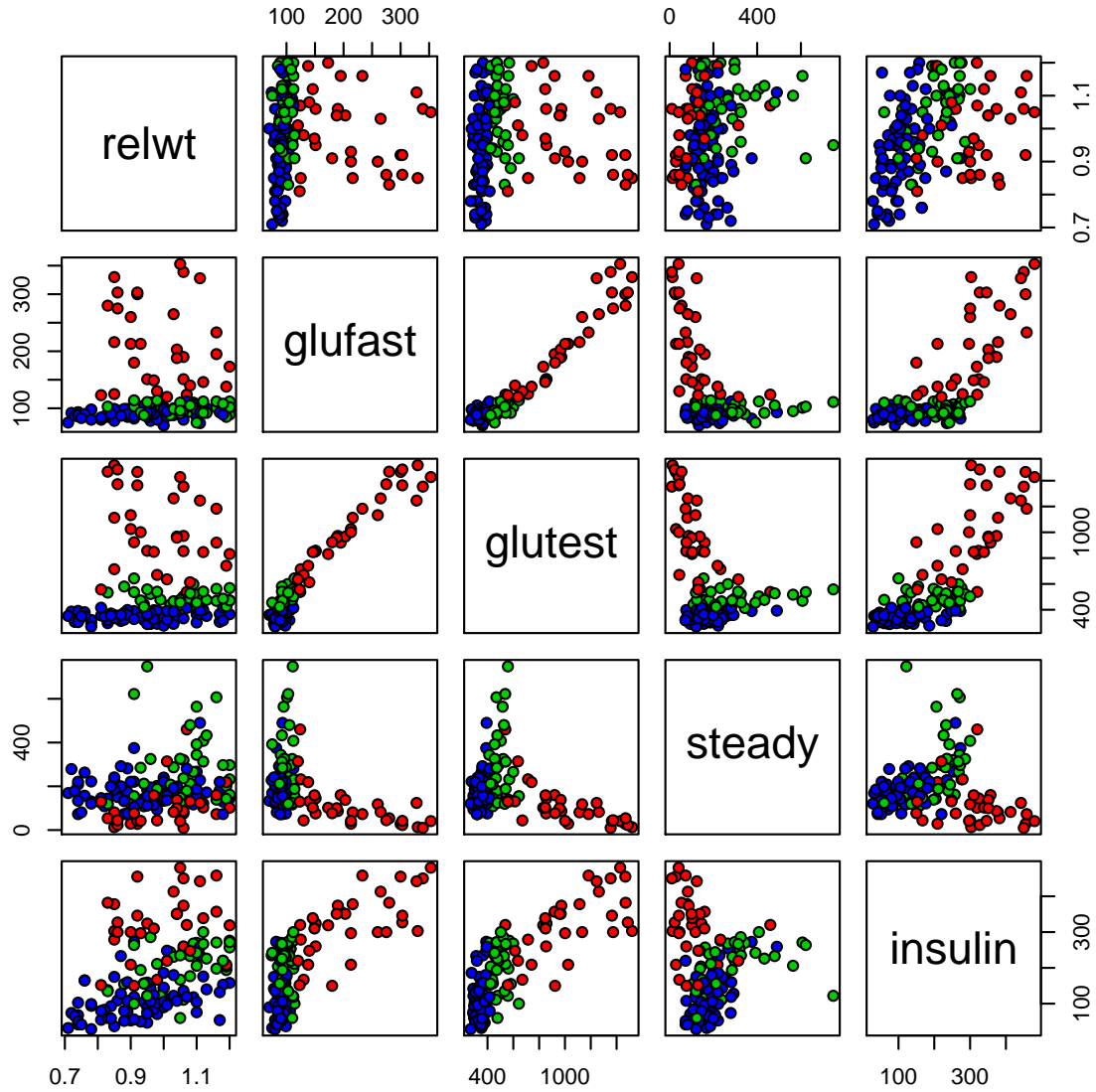
QR Column 101



QR Column 501



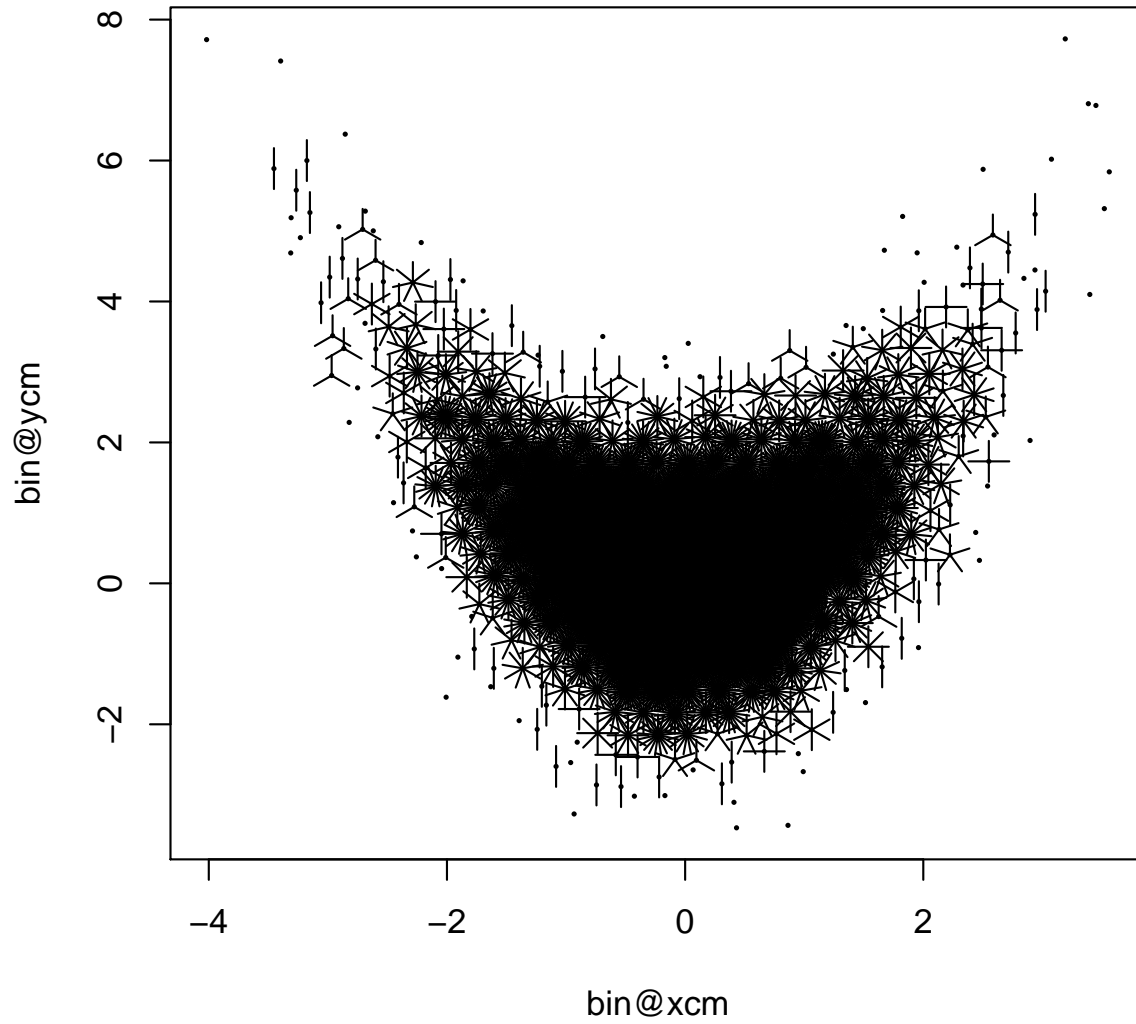
Scatterplots



- Pair Plots

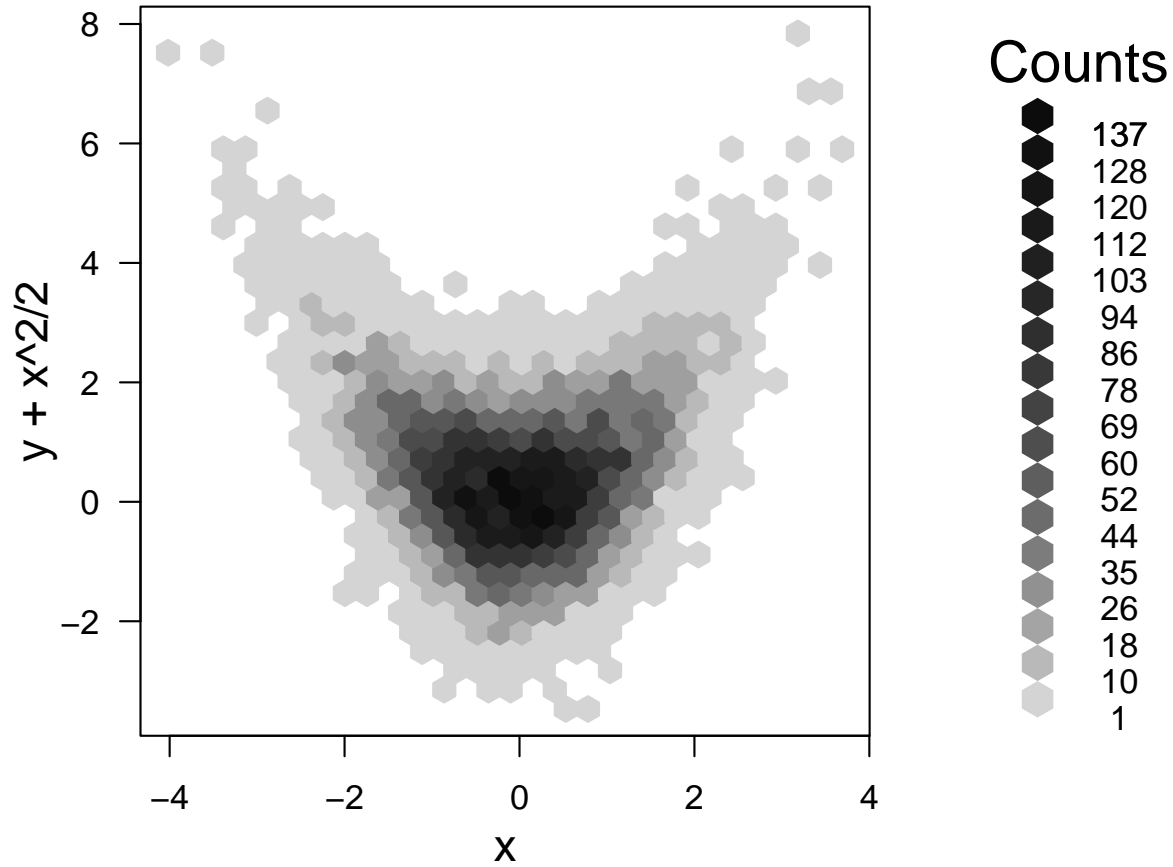
● Sunflower Plots

$(X, X*X/2 + Y)$ where $X, Y \sim \text{rnorm}(10000)$



● Hexagonal Binning

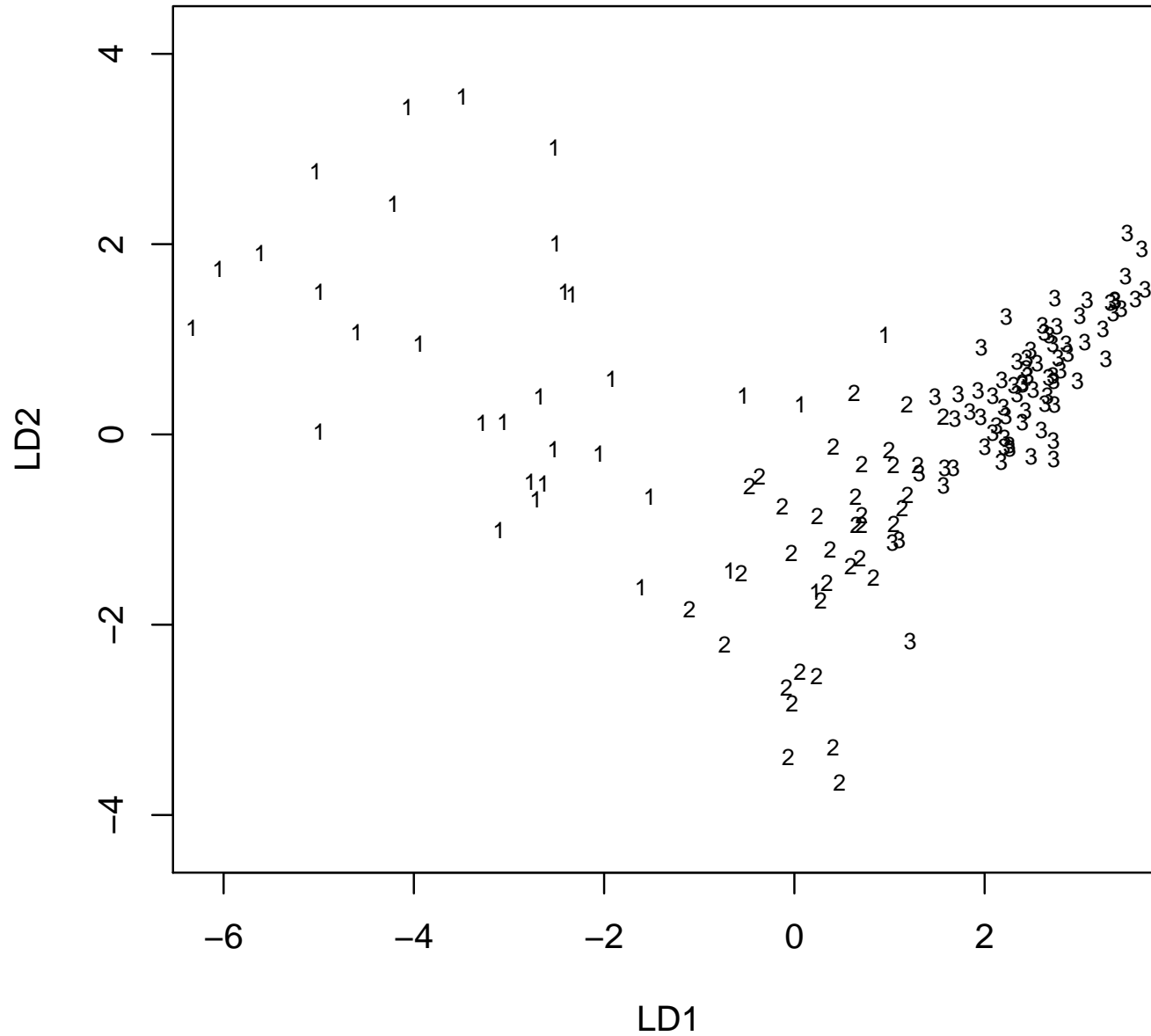
$(X, X^2/2 + Y)$ where $X, Y \sim \text{rnorm}(10000)$



Special Transformations of Variables

- Maximize intergroup variance:

Linear Discriminant Analysis.



(1='Overt Diabetic' , 2='Chem. Diabetic')

- Maximize Chi-square :

Correspondence Analysis (Finding an underlying ranking.)

Example: Cox and Brandwood tried to seriate Plato's works using discriminant analysis on the proportion of sentence endings in a given book, with a given stress pattern. Here we show how such an analysis can be done with correspondence analysis on the table of frequencies of sentence endings

An Example

Correspondence analysis is useful when all the variables have the same status. This is sometimes called unsupervised learning, and includes clustering, principal components as well. They have in common the creation of a new set of variables that simplify the arrays at hand. In the case of clustering, the new variable is a categorical, in correspondence analysis and principal components the new variables are continuous and enable the construction of useful new graphical representations of the data.

Correspondence analysis is an exploratory method because it does not presuppose any model for the data, as do Goodman's bilinear methods or factor analysis models for instance. Correspondence analysis and principal components can both be extended to three-way arrays, for instance for the analysis of bootstrap permutation tests or time series of matrices. Such data are often called data cubes.

Correspondence Analysis

Correspondence analysis (CA, also called homogeneity analysis and reciprocal averaging), can be used to analyse several types of multivariate data. All involve some categorical variables. Here are some examples of the type of data that can be decomposed using this method:

- Contingency Tables (cross between two categorical variables)
- Multiple Contingency Tables (cross between several categorical variables).
- Binary tables obtained by cutting continuous variables into classes and then recoding both these variables and any extra categorical variables into 0/1 tables, 1 indicating presence in that class. So for instance a continuous variable cut into three classes will provide three new binary variables of which only one can take the value 1 for any given observation.

To first approximation, correspondence analysis can be understood as an

extension of principal components analysis (PCA) where the variance in PCA is replaced by an inertia proportional to the χ^2 distance of the table from independence. CA decomposes this measure of departure from independence along axes that are orthogonal according to the χ^2 inner product. If we are comparing two categorical variables, the simplest possible model is that of independence in which case the counts in the table would obey approximately the margin products identity for a $m \times p$ contingency table with a total sample size of $n = \sum_{i=1}^m \sum_{j=1}^p n_{ij} = n \dots$

Independence

$$n_{ij} \doteq \frac{n_{i.} n_{.j}}{n} n$$

can also be written: $\mathbf{N} \doteq \mathbf{c} \mathbf{r}' \mathbf{n}$, where

$$\mathbf{c} = \frac{1}{n} \mathbf{N} \mathbf{1}_m \quad \text{and} \quad \mathbf{r}' = \frac{1}{n} \mathbf{N}' \mathbf{1}_p$$

The departure from independence is measured by the χ^2 statistic

$$\chi^2 = \sum_{i,j} \left[\frac{(n_{ij} - \frac{n_{i.} n_{.j}}{n} n)^2}{\frac{n_{i.} n_{.j}}{n} n} \right]$$

Example: Eye color -Hair color

Here is a simple contingency table as our example from Snee (1974)[10].

eyes	Black	Brunette	Red	Blonde
Brown	68	20	15	5
Blue	119	84	54	29
Hazel	26	17	14	14
Green	7	94	10	16

This summarizes the data:

Black	Brunette	Red	Blonde	Brown	Blue	Hazel	Green
1	0	0	0	1	0	0	0
1	0	0	0	1	0	0	0
1	0	0	0	1	0	0	0
1	0	0	0	1	0	0	0
⋮	⋮	⋮			
1	0	0	0	0	0	0	1
⋮	⋮	⋮			

```
> chisq.test(eyes)
      Pearson's Chi-squared test
data:  eyes, X-squared = 138.2898,
df = 9, p-value = < 2.2e-16
```

This is a very extreme point in the χ^2 distribution. The inertia is the χ -squared statistic divided by the number of observations (`sum(eyes)`); Here, the inertia is $138.3/592 = 0.2336$. CA decomposes this inertia into the sum of eigenvalues of a symmetrized reweighted version of the original table, as will be explained below.

The R command `resca<-ca.us(eyes)`

provides first a screeplot of these eigenvalues, in this example:

	Values	Percent	
1	0.2088	89.	*****
2	0.0222	10.	**
3	0.0026	1.	.
Total	0.2336	100.	

```
> resca=ca.us(eyesc)
```

```
  Eigenvalue inertia % cumulative %
1      0.2090      89.37      89.37
2      0.0222       9.51      98.88
3      0.0026       1.11      99.99
```

Please examine the histogram of the eigenvalues...

How many axes do you wish to be retained? (<= 3)

1: 3

Read 1 items

COLUMNS

(cta = cta * 1000

ctr = ctr * 1000)

	Axis	1		Axis	2		Axis	3	
	coord.	cta	ctr	coord.	cta	ctr	coord.	cta	ctr
Black	-0.49	431	967	0.088	130	31	-0.0220	67	2
Brunette	0.55	521	977	0.083	112	22	0.0047	3	0
Red	-0.21	34	542	-0.170	198	336	0.1000	611	121
Blonde	0.16	14	176	-0.340	559	773	-0.0880	319	52

ROWS

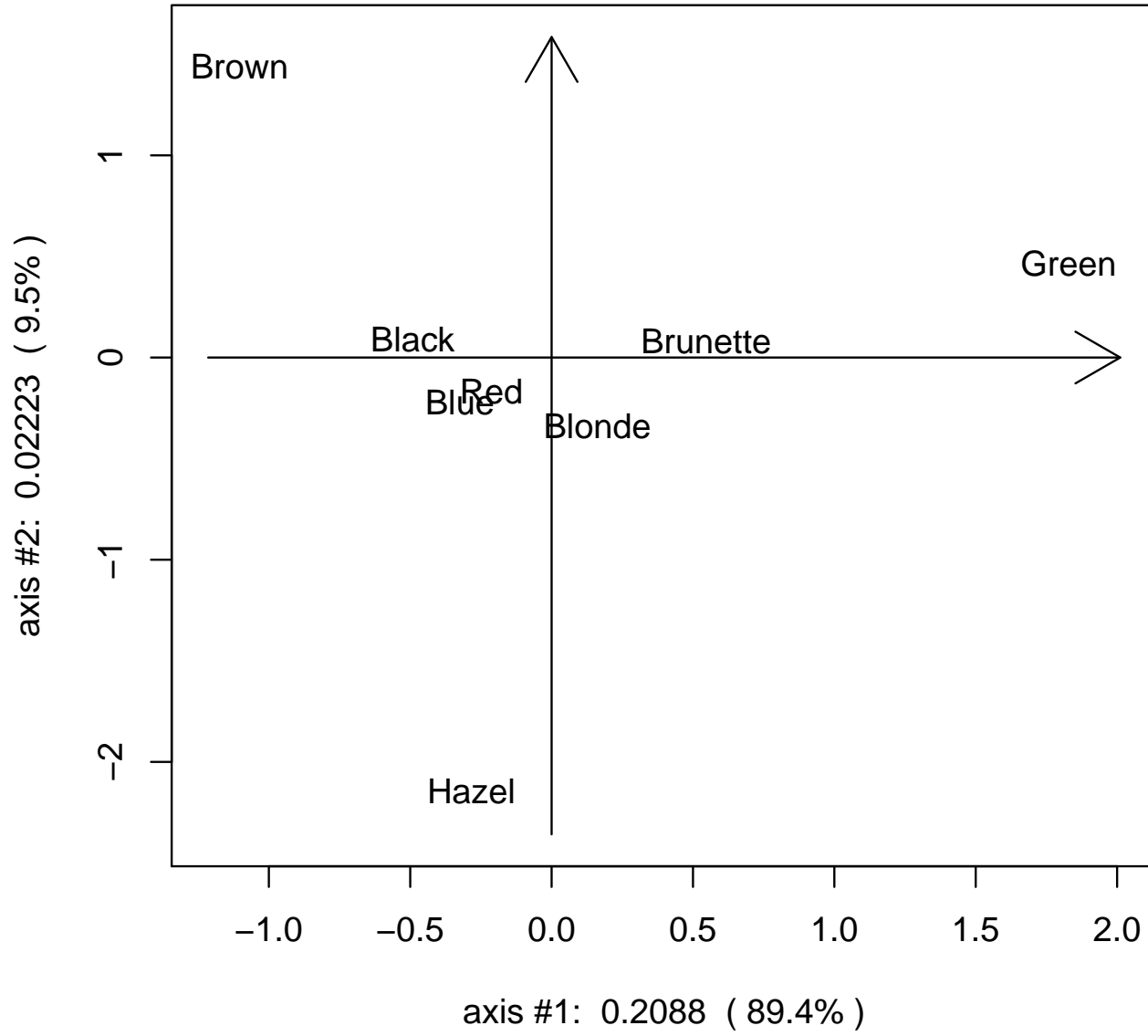
(cta = cta * 1000

ctr = ctr * 1000)

	Axis	1		Axis	2		Axis	3	
	coord.	cta	ctr	coord.	cta	ctr	coord.	cta	ctr
Brown	-0.50	222	838	0.210	379	152	-0.056	216	10
Blue	-0.15	51	864	-0.033	23	42	0.049	443	94
Hazel	-0.13	10	133	-0.320	551	812	-0.083	319	55

Green 0.84 717 993 0.070 47 7 -0.016 22 0

Correspondance Analysis : columns as centroids of the rows



R output from caplot

Formulation as a generalized singular value decomposition

Given an $m \times p$ contingency table of counts \mathbf{N} of m levels for a row variable and p levels for a column variable. (This is equivalent to a binary matrix X with $n = \sum_{ij} n_{ij} = n_{..}$ observations on $m + p$ columns, a notion that is useful of the generalisation later.)

The first transformation makes the contingency matrix \mathbf{N} into a frequency matrix $\mathbf{F} = \frac{1}{n}\mathbf{N}$. We will denote the row sums by $\mathbf{r} = \mathbf{F}\mathbf{1}_p$ and the column sums by the vector $\mathbf{c} = \mathbf{F}'\mathbf{1}_m$. These both sum to one

$$\mathbf{r}'\mathbf{1}_m = \mathbf{c}'\mathbf{1}_p = 1$$

In the case of independence

$$\mathbf{F} \doteq \mathbf{r}\mathbf{c}'$$

All the rows would be multiples of each other or as this is sometimes

called, *homogeneous*. So, if all the rows were divided by the weight of that row, these so-called *row profiles* $\mathbf{F}\mathbf{D}_r^{-1}$ would be equal ($\mathbf{F}\mathbf{D}_r^{-1} = \mathbf{1}_m\mathbf{c}$), where \mathbf{D}_r^{-1} denotes the diagonal matrix with the vector \mathbf{r}^{-1} on its diagonal.

The average row in the case of homogeneity and independence is obtained by averaging the rows with the relevant weights for each column. The average of the row-profiles is \mathbf{c} . The departure from independence and homogeneity is measured by some norm of $\mathbf{F}\mathbf{D}_r^{-1} - \mathbf{1}_m\mathbf{c}$ (or at the term by term level $\frac{f_{ij}}{f_{i.}} - f_{.j}$). With this notation we remark that

$$\begin{aligned}\chi^2 &= n \sum_{i,j} \frac{(f_{ij} - r_i c_j)^2}{r_i c_j} \\ &= n \sum_{i,j} r_i c_j \left(\frac{f_{ij}}{r_i c_j} - 1 \right)^2\end{aligned}$$

Verification in **R**:

```
> F<-eyes/sum(eyes)
> r<-apply(F,1,sum)
> c<-apply(F,2,sum)
> E<-outer(r,c)
> sum((F-E)^2/E)*592
[1] 138.2898
```

To compute the distance between profiles, each column is reweighted by the inverse of its sum, this gives the χ^2 distance between row profiles.

$$\begin{aligned}\chi^2 &= n \operatorname{trace} ((\mathbf{F} - \mathbf{rc}')' \mathbf{D}_r^{-1} (\mathbf{F} - \mathbf{rc}') \mathbf{D}_c^{-1}) \\ &= \operatorname{trace} (\mathbf{A}' \mathbf{A}) \quad \text{where } \mathbf{A} = \mathbf{D}_{\sqrt{r}}^{-1} (\mathbf{F} - \mathbf{rc}') \mathbf{D}_{\sqrt{c}}^{-1}\end{aligned}$$

The latter decomposition shows a justification for choosing the matrix \mathbf{A} as a natural square root. $\mathbf{W} = \mathbf{A}' \mathbf{A}$ is in a sense the characteristic matrix-operator of the analysis, in the same way the covariance or correlation matrices are those of principal components analysis.

Generalising principal component analysis to include metrics on the rows and the columns can lead to other multivariate techniques such as discriminant analysis (See Mardia, Kent and Bibby(1979) [9]).

Correspondence analysis decomposes the matrix \mathbf{W} : its eigenvectors give the axes that account for the largest part of the departure from independence, just as principal components provides the axes accounting for the largest variability. Computationally this is achieved by a generalized singular value decomposition

$$\mathbf{D}_r^{-1}\mathbf{F}\mathbf{D}_c^{-1} - \mathbf{1}'_m\mathbf{1}_p = \mathbf{U}\mathbf{S}\mathbf{V}',$$

with $\mathbf{V}'\mathbf{D}_c\mathbf{V} = \mathbf{I}_p, \mathbf{U}'\mathbf{D}_r\mathbf{U} = \mathbf{I}_m$

equivalent to the eigendecomposition $\mathbf{W} = \mathbf{A}'\mathbf{A} = \mathbf{V}'\mathbf{S}^2\mathbf{V}$ or the singular value decomposition

$$\mathbf{D}_r^{-\frac{1}{2}}\mathbf{F}\mathbf{D}_c^{-\frac{1}{2}} - \sqrt{\mathbf{r}}\sqrt{\mathbf{c}}' = (\mathbf{D}_r^{\frac{1}{2}}\mathbf{U})\mathbf{S}(\mathbf{D}_c^{\frac{1}{2}}\mathbf{V})',$$

where $(\mathbf{D}_c^{\frac{1}{2}}\mathbf{V})'(\mathbf{D}_c^{\frac{1}{2}}\mathbf{V}) = \mathbf{I}_p$, and $(\mathbf{D}_r^{\frac{1}{2}}\mathbf{U})'(\mathbf{D}_r^{\frac{1}{2}}\mathbf{U}) = \mathbf{I}_p$.

Reminder : Eigenvalues and Singular Values

Description of singular value Decomposition

This is the most important matrix decomposition in statistics.

```
> u=as.matrix(c(3, 1, -1, 2))
```

```
      [,1]
```

```
[1,]    3
```

```
[2,]    1
```

```
[3,]   -1
```

```
[4,]    2
```

```
> v=(1:4)
```

```
      [1] 1 2 3 4
```

```
> X=u%*%t(v)
```

```
> X
```

```
      [,1] [,2] [,3] [,4]
```

```
[1,]    3    6    9   12
```

```
[2,]    1    2    3    4
```

```
[3,]   -1   -2   -3   -4
```

```
[4,]    2    4    6    8
```

```
>svd(X)
```

```
$d
```

```
[1] 2.121320e+01 5.002854e-16 5.260193e-48 3.861862e-80
```

```
$u
```

	[,1]	[,2]	[,3]	[,4]
[1,]	-0.7745967	0.6324555	5.312218e-17	1.046073e-17
[2,]	-0.2581989	-0.3162278	-9.128709e-01	-1.760937e-16
[3,]	0.2581989	0.3162278	-1.825742e-01	8.944272e-01
[4,]	-0.5163978	-0.6324555	3.651484e-01	4.472136e-01

```
$v
```

	[,1]	[,2]	[,3]	[,4]
[1,]	-0.1825742	0.98319208	0.0000000	0.000000e+00
[2,]	-0.3651484	-0.06780635	-0.9284767	-3.171855e-17
[3,]	-0.5477226	-0.10170953	0.2228344	8.000000e-01
[4,]	-0.7302967	-0.13561270	0.2971125	-6.000000e-01

```
> E=10^(-3)*matrix(rnorm(16),4,4);
```

```
>> XE=X+E
```

```
XE =
```

```
          [,1]      [,2]      [,3]      [,4]
[1,]  2.99960  6.00034  8.99998 11.99909
[2,]  0.99874  2.00057  2.99935  3.99880
[3,] -0.99966 -1.99969 -2.99881 -3.99944
[4,]  2.00112  3.99919  6.00021  7.99984
```

```
> round(svd(XE)$d,5)
```

```
[1] 21.21203  0.00209  0.00076  0.00001
```

#An example you can't see with your bare eyes:

```
> v=v+0.01*rnorm(4)
```

```
> u=u+0.01*rnorm(4)
```

```
> X2=u%*%t(v)
```

```
> X2
```

```
      [,1]      [,2]      [,3]      [,4]
[1,]  3.076361  6.026556  9.003151 11.991022
[2,]  1.005600  1.969960  2.942949  3.919623
[3,] -1.029503 -2.016785 -3.012901 -4.012791
[4,]  2.030531  3.977786  5.942467  7.914589
```

```
> round(svd(X2)$d,4)
[1] 21.1601  0.0000  0.0000  0.0000
```

Here is what we need to remember:

$$X = USV', V'V = I, U'U = I, S \text{ diagonal } s_i$$

Actually the singular values are the square roots of the eigenvalues of $X'X$.

The CA plots can be used to find out if there is an hidden ordination of the data, as for instance the chronological seriation studied below.

Finding an underlying ranking.

Example: Cox and Brandwood [3] tried to seriate Plato's works using discriminant analysis on the proportion of sentence endings in a given book, with a given stress pattern. Here we show how such an analysis can be done with correspondence analysis on the table of frequencies of sentence endings¹. The first 10 profiles (as percentages) look as follows:

	Rep	Laws	Crit	Phil	Pol	Soph	Tim
UUUUU	1.1	2.4	3.3	2.5	1.7	2.8	2.4
-UUUU	1.6	3.8	2.0	2.8	2.5	3.6	3.9
U-UUU	1.7	1.9	2.0	2.1	3.1	3.4	6.0
UU-UU	1.9	2.6	1.3	2.6	2.6	2.6	1.8
UUU-U	2.1	3.0	6.7	4.0	3.3	2.4	3.4
UUUU-	2.0	3.8	4.0	4.8	2.9	2.5	3.5
--UUU	2.1	2.7	3.3	4.3	3.3	3.3	3.4

¹A computerized analysis of Plato's work appears in Ledger (1989)[8]

```
-U-UU 2.2  1.8  2.0  1.5  2.3  4.0  3.4
-UU-U 2.8  0.6  1.3  0.7  0.4  2.1  1.7
-UUU- 4.6  8.8  6.0  6.5  4.0  2.3  3.3
.....etc (there are 32 rows in all)
```

The eigenvalue decomposition (called the scree plot) shows that two axes will provide a summary of 85% of the departure from independence.

```
> res.plato=ca.us(platon)
Eigenvalue inertia % cumulative %
1      0.09170      68.96      68.96
2      0.02120     15.94     84.90
3      0.00911      6.86     91.76
4      0.00603      4.53     96.29
5      0.00276      2.07     98.36
6      0.00217      1.64    100.00
```

Please examine the eigenvalues...

How many axes do you wish to be retained?

The function `ca.us` asks this question interactively, because only the screeplot visualisation can protect against the separation of 2 or more very close eigenvalues. The function then returns as the resulting data list the coordinates for plotting that can be visualized using `ggobi[?]` if there are more than 3 relevant axes. Here is the correspondence analysis representation for the rows and columns taken separately. We have made the labels' sizes proportional to the quality of the representations in this first plane. These are obtained by typing `caplot.us(res.plato)`.

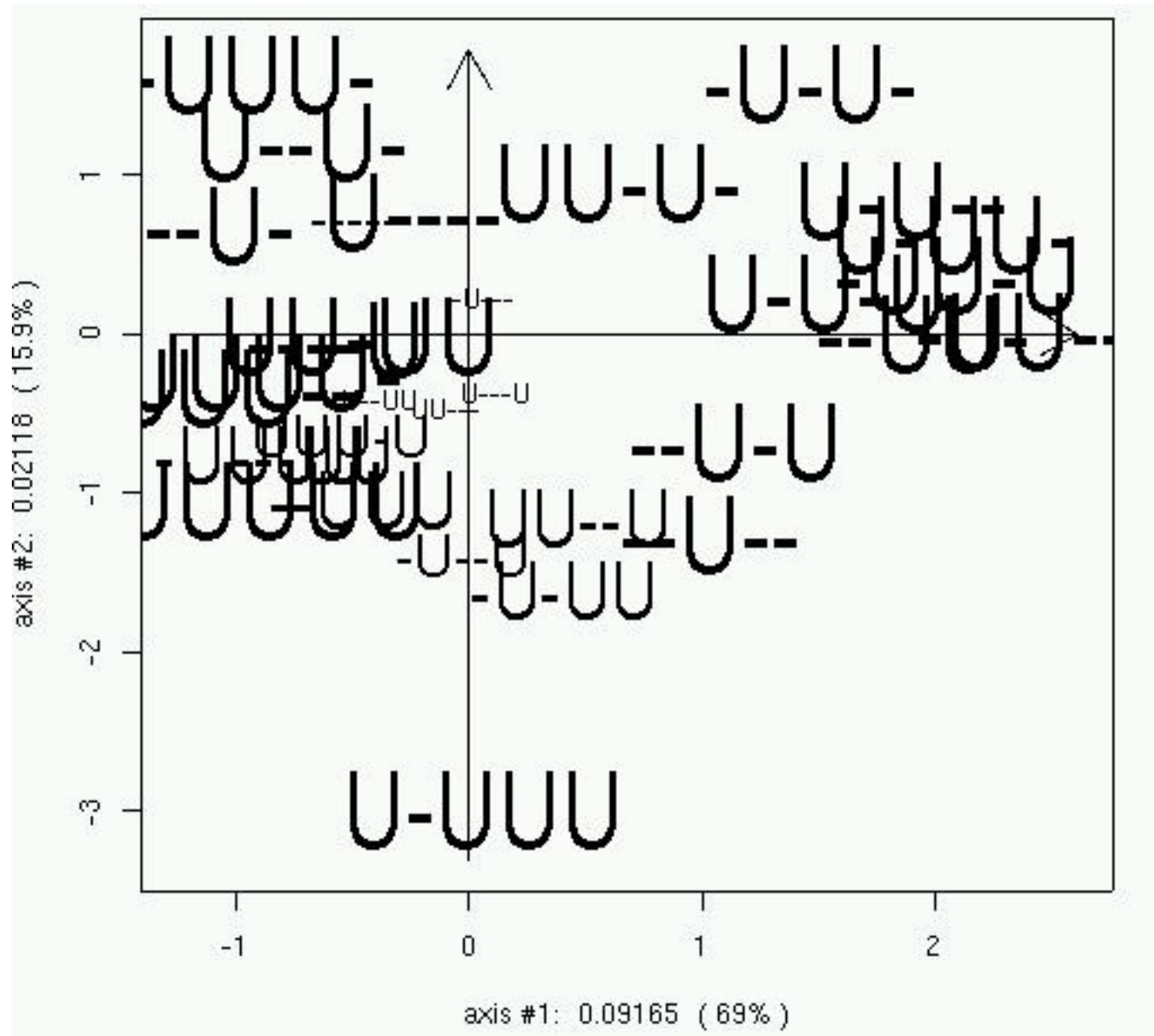


Figure 2: Plato sentence endings (rows)

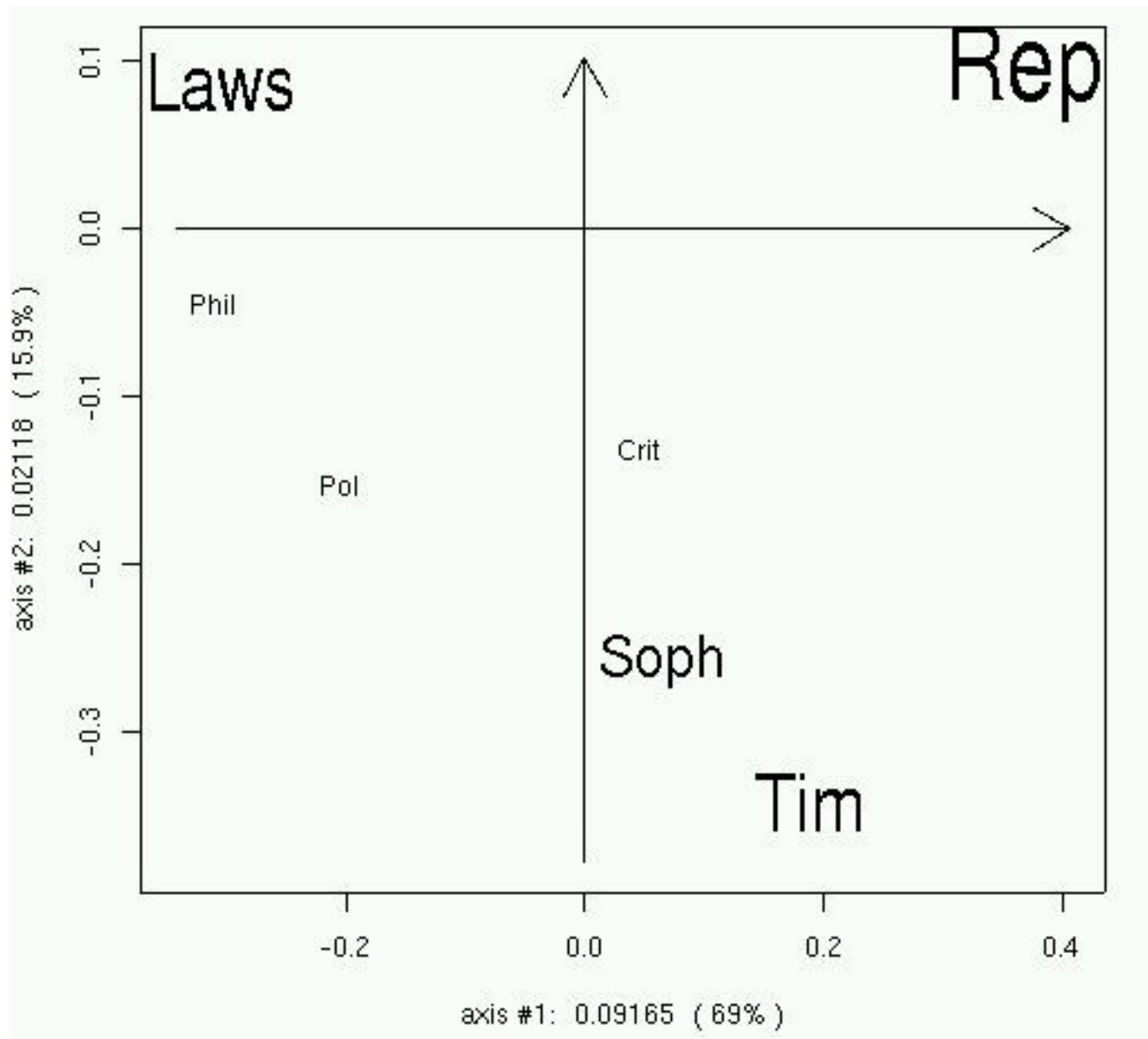


Figure 3: Plato's works (columns)

The plot shows very clearly a seriation of all the works, in fact, except for <http://encyclopediaindex.com/b/criti10.htm> **Critias**, the seriation is determined by the first axis, the second axis helps to place **Critias** between **Politicus** and **Sophist**, a choice that has also been validated in Ledger(1989) and Cox and Brandwood (1959).

Decomposition of Inertia

χ^2 distance between profiles

Here is a reason why such a weighted distance could be useful : Take a very simple contingency table :

$$X = \begin{bmatrix} 6 & 2 & 12 \\ 15 & 5 & 7 \\ 21 & 7 & 42 \end{bmatrix}$$

The row profiles are all

$$\begin{array}{l} \quad \quad \quad [,1] \quad [,2] \quad [,3] \\ [1,] \quad 0.3 \quad 0.1 \quad 0.6 \\ [2,] \quad 0.3 \quad 0.1 \quad 0.6 \\ [3,] \quad 0.3 \quad 0.1 \quad 0.6 \end{array}$$

We can see that the profiles are identical, the multinomials generating the rows are said to be homogenous, and the contingency table is in fact only of rank 1:

$$X = \begin{bmatrix} 2 \\ 5 \\ 7 \end{bmatrix} \%*\% \begin{bmatrix} 3 & 1 & 6 \end{bmatrix}$$

This is exactly the problem we encounter in principal components analysis, we need the decomposition in singular values of X , but instead of centering the data with regards to the mean, the data will be centered at independence. Another difference lies with the choice of the metric for computing distances between rows or columns.

Decomposition of the difference from independence

A cloud, or scatter of weighted points :

These are points defined in an euclidean space, say \mathbb{R}^p for instance, so that distances between them are easy to compute. However we associate to each multidimensionnal point a weight that changes the inertia of the scatterpoints. For instance if we have two points the one with a higher weight will 'pull ' the centre of gravity towards it. The same will happen for the 'minimum inertia ' line, it will be pulled towards highly weighted points.

Distributional Equivalence :

If we add two rows that have the same profiles, this will not change the axes chosen to represent the data, (the column profiles' geometry remains unchanged). Thinking of the points as weighted points in a cloud, two points that would be at the same spot can be merged because we can add their weights.

Barycentric Representation :

Take the simplest case: row profiles of a 3-column contingency table. The profiles sum to one so are all representable in a triangle (called the 3-dimensional simplex). The vertices are the extreme profiles, say $(1,0,0)$, $(0,1,0)$ and $(0,0,1)$. Although the row profiles are in a three dimensional space as they belong to this triangle they can be taken out and just looked at in these coordinates, called the barycentric co-ordinate system. Now an extra scale change will bring this representation to the correspondence analysis one : the dimensions will be weighted inversely by the relative weights of the columns , called column mean profiles, (which also add to 1). The distances we want to represent between points are to be the χ^2 distances relevant here, so the sides of the equilateral triangle are stretched to have sides inversely proportional to the square roots of their mean values.

The side of the triangle the most stretched corresponds to the least frequent column.

This representation is the one chosen by default by the function `caplot`,

there is a delicate issue of choosing the scales in the two dimensions so that simultaneous representations of rows and columns are valid (the program warns the user when such a scaling has not been chosen). In the relevant choice of scaling, proximities between row and column points are hard to interpret, however it is easier to interpret the directions of the different rows and columns.

Reading the Output

Although the maps provided by doing both correspondence and principal components analysis look quite simple there are traps that lead to *misinterpretations* that must be avoided. Associated to the co-ordinates in the new spaces are what we call *loadings* or *contributions* and which are indicators of how true the proximities in the image space are. To this end the object that the function `caplot` produce a listed output containing the eigenvalues, coordinates for the rows, for the columns, that are used for building the graphical representations and absolute contributions and squared cosines that are important diagnostic tools.

```
>names(resca)
[1]"recap" "valp" "ind.cords" "var.cords" "ind.cta"
[6]"var.cta" "ind.ctr" "var.ctr" "ind" "var" "axes"
```

- When trying to understand the most meaningful rows or columns for a given axis we look at the absolute contributions of rows or columns to given axis, this gives the amount of an axis's inertia explained by single row or column.
- The relative contribution of an axis/ of two axes to the inertia of a row This is the same as the cosine of the point with the axis that says how

well a point is being projected onto the axis.

Contribution² to the inertia from row i :

Distance from the i th row to the center of the row-points:

$$\begin{aligned}d_{\chi^2}^2(\text{profile}_i, \text{center}) &= \sum_j \frac{1}{f_{\cdot j}} \left(\frac{f_{ij}}{f_i} - f_{\cdot j} \right)^2 \\ &= \sum_j f_{\cdot j} \left(\frac{f_{ij}}{f_i \cdot f_{\cdot j}} - 1 \right)^2 = \sum_k (s_k u_i^k)^2\end{aligned}$$

This row will thus participate to the inertia by this amount weighted by the row's mass r_i . This can be decomposed into each of the axis separately thus giving an idea of the contribution of each row to the inertia of each axis, this is called the absolute contribution of row i to axis k , $r_i \sum_k (s_k u_i^k)^2$. The sum of all row's contributions to a given axis add to one. This translates the fact that $\mathbf{U}'\mathbf{D}_r\mathbf{U} = \mathbf{I}_m$, here are the absolute contributions for the eyes data:

²Sometimes called absolute contribution as different from the \cos^2 which are sometimes called relative contributions.

```
%Output resca$indcta
      Axis 1      Axis 2      Axis 3
Brown  0.2225    0.3788    0.2163
Blue   0.0509    0.0232    0.4428
Hazel  0.0096    0.5513    0.3191
Green  0.7170    0.0467    0.0217
Total  1.        1.        1.
```

Thus we can see that the most important row category for explaining the first axis is Green eyes. The contributions to inertia as decomposed according to the column categories gives:

```
%Output resca$varcta
      Axis 1      Axis 2      Axis 3
Black  0.4312    0.1304    0.0668
Brunette 0.5213    0.1124    0.0031
Red    0.0340    0.1980    0.6109
Blonde 0.0135    0.5591    0.3192
Total  1.        1.        1.
```

Squared Cosine

For interpretation of the exactitude of the projection, it is important to consult the cosine of the angle between the point and its projection onto the k th axis or the plane or space spanned by the relevant axes.

$$\cos^2(\text{row } i, \text{axe } k) = \frac{(s_k u_{ik})^2}{\sum_k (s_k u_{ik})^2}$$

%Output resca\$indctr

	Axis 1	Axis 2	Axis 3	Total
Brown	0.838	0.152	0.010	1.
Blue	0.864	0.042	0.094	1.
Hazel	0.133	0.812	0.055	1.
Green	0.993	0.007	0.000	1.

In this case of course, taking 3 axes results in a complete reconstruction of the table, so the \cos^2 between the rows and this 3-space is 1.

On the other hand the column's cosine can also be provided, here the output gives:

```
%Output  resca$varctr
          Axis 1  Axis 2  Axis 3  Total
Black    0.9670  0.031  0.002   1.
Brunette 0.9775  0.022  0.000   1.
Red      0.5424  0.336  0.121   1.
Blonde   0.1759  0.777  0.052   1.
```

This information can actually be incorporated into the graphic by making the size of the points label proportional to how close the point is from the plane. This is a 'perspective' type plot, points which are close, are well represented and have high cosines. Thus the large letters label points that are close to the plane, and are thus "well-represented".

Summing these for the 2 for which the representation was made, and using these indices as the size of the font, enables one to see at a glance which rows and columns are important to interpret.

Computations

```
eyes = matrix( c(68,119,26,      7,
                20,  84  ,  17  ,  94,
                15  ,  54  ,  14  ,  10,
                5   ,  29  ,  14  ,  16 ),byrow=T,ncol=4)
eyesc=c('Black','Brunette','Red','Blonde')
eyesn=c('Brown','Blue','Hazel','Green')
sum(eyes)
      592
F=eyes/592
F =
      0.1149      0.2010      0.0439      0.0118
      0.0338      0.1419      0.0287      0.1588
      0.0253      0.0912      0.0236      0.0169
      0.0084      0.0490      0.0236      0.0270
one4=rep(1,4)
```

```
r=F%*%one4
```

```
0.3716
```

```
0.3632
```

```
0.1571
```

```
0.1081
```

```
c=t(F)%*%one4
```

```
0.1824
```

```
0.4831
```

```
0.1199
```

```
0.2145
```

```
indep=round(592*r%*%t(c))
```

```
40    106    26    47
```

```
39    104    26    46
```

```
17     45    11    20
```

```
12     31     8    14
```

```
diag(1/(apply(t(indep),2,sum))%*%indep
```

```
0.1824    0.4831    0.1199    0.2145
```

```
0.1824    0.4831    0.1199    0.2145
```

```

0.1824    0.4831    0.1199    0.2145
0.1824    0.4831    0.1199    0.2145
indep%*%diag(1/apply(indep,2,sum))
0.3716    0.3716    0.3716    0.3716
0.3632    0.3632    0.3632    0.3632
0.1571    0.1571    0.1571    0.1571
0.1081    0.1081    0.1081    0.1081
resid=eyes-indep
27.8649   12.7162   -0.3851  -40.1959
-19.2230 -19.8682   -8.7855   47.8767
-1.9662    9.0709    2.8463   -9.9510
-6.6757   -1.9189    6.3243    2.2703

> diag(as.vector(1/sqrt(r)))
      [,1]      [,2]      [,3]      [,4]
[1,] 1.640399 0.000000 0.000000 0.000000
[2,] 0.000000 1.659364 0.000000 0.000000
[3,] 0.000000 0.000000 2.523012 0.000000

```

```
[4,] 0.000000 0.000000 0.000000 3.041381
```

```
a=diag(as.vector(1/sqrt(r)))*%resid%*%diag(as.vector(1/sqrt(r)))
```

```
> sum(diag(t(a)*%a))
```

```
[1] 138.2898
```

Using ade4 for CA

```
dimnames(eyes)=list(eyesn,eyesc)
eyes.df=data.frame(eyes)
eyes.coa <- dudi.coa(eyes.df, scannf = TRUE)
2
par(mfrow = c(1,2))
s.label(eyes.coa$co, clab = 0.6)
s.label(eyes.coa$li, clab = 0.6)
```

Codon Usage Data

(see <http://codonw.sourceforge.net/culong.html>)

The vast majority of prokaryotic and eukaryotic species have non-random codon usage. The major factor in codon choice in many unicellular and some multicellular organisms is Darwinian selection between synonyms; highly expressed genes using a restricted set of codons (Gouy and Gautier 1982; Ikemura 1985; Sharp and Matassi 1994). This selection is almost certainly for optimal translational efficiency, and is most pronounced in highly expressed genes in species whose effective population size is large (Bulmer 1991; Li, 1987). Divergence of codon usage and choice of optimal codons correlates with evolutionary distance, but usage patterns in phylogenetically distant species may converge due to the similarities of factors that influence the drift in choice of optimal codons.

For *E. coli* and yeast genes, the main evolutionary force varying among genes was considered to be natural selection to optimise protein production (translational selection, acting on highly expressed genes), whereas with human genes, it was thought to be variation among chromosomal regions in the mutation process (biased mutation, resulting in base composition differences).

Analysis of codon usage has been used to identify highly expressed genes. (Cancilla et al. 1995b; Freirepicos et al. 1994; Gharbia et al. 1995). Atypical codon usage has been used to infer that genes have been acquired by horizontal transfer (Delorme et al. 1994; Groisman et al. 1992; Medigue et al. 1991).

ade4 Package

```
#####  
#  
# Load required packages, here ade4 for multivariate analyses and seqinR  
# to access ACNUC databases. This is dispensable since these packages  
# are automatically pre-loaded in our RWeb server, but would be mandatory  
# for running the script in a different context.  
#  
#####  
  
library(ade4)  
library(seqinr)  
  
#####  
#  
# Choose a bank, here the bank trypano is a frozen subset of GenBank:  
# its content is stable to allow for the reproducibility of results.  
# It was build on 27-JAN-2004 from GenBank release 139 by selecting  
# data from Leishamnia major, Trypanosoma brucei and Trypanosoma cruzi.  
# It contains 117,177,046 bases 158,838 sequences 4,744 subsequences  
# and 2,114 references.  
#  
#####  
  
choosebank(bank = "trypano")  
  
#####  
#  
# Search the bank for complete nuclear coding sequence for our three species:  
#  
#####
```

```
myquery <- function(listname, species)
{
  requested <- paste("SP=", species, "ET O=nuclear ET T=cds ET NO K=partial")
  query(listname, requested)
}
```

```
myquery("lm", "Leishmania major")
myquery("tb", "Trypanosoma brucei")
myquery("tc", "Trypanosoma cruzi")
```

```
#####
#
# The mnemonics of corresponding sequences are stored in lm$req (1,467
# sequences) tb$req (1,772 sequences) and tc$req (679 sequences). From
# these we can now retrieve the sequences:
#
#####
```

```
seqlm <- lapply(lm$req, getseq, as.string = TRUE)
seqtb <- lapply(tb$req, getseq, as.string = TRUE)
seqtc <- lapply(tc$req, getseq, as.string = TRUE)
```

```
#####
#
# From sequences, build dataframes with codon usage for each sequence.
#
#####
```

```
codons <- words(length = 3, alphabet = s2c("acgt"))
```

```
myuco <- function(seq)
{
  start <- seq(1, nchar(seq)-2, by = 3)
  as.vector(table(factor(substring(seq, start, start+2), codons)))
}
```

```
mkdata <- function(seqs)
{
  tab <- sapply(seqs, myuco)
  tab <- as.data.frame(tab, row.names = codons)
  return( tab )
}
```

```
tablm <- mkdata(seq1m)
tabtb <- mkdata(seqtb)
tabtc <- mkdata(seqtc)
```

```
#####
```

```
#
```

```
# Remove CDS with in-frame stop codons (translated by "*")
```

```
# and CDS that have less than 100 codons:
```

```
#
```

```
#####
```

```
cleanup <- function(tab)
```

```
{
```

```
  aa <- translate(sapply(rownames(tab), s2c))
```

```
  tab <- tab[ , colSums(tab[which(aa == "*"), ]) == 1]
```

```
  tab <- tab[ , colSums(tab) > 100 ]
```

```
    return(tab)
}

tablm <- cleanup(tablm) # now 1,427 cds
tabtb <- cleanup(tabtb) # now 1,298 cds
tabtc <- cleanup(tabtc) # now    635 cds

#####
#
# Run correspondence analysis on merged dataset:
#
#####

tab <- cbind(tablm, tabtb, tabtc)
names(tab) <- 1:ncol(tab)
coa <- dudi.coa(tab, scan = FALSE, nf = 2)

#####
```

```
#
# Run synonymous codon usage analysis:
#
#####

facaa <- as.factor(aaa(translate(sapply(rownames(tab), s2c)
scua <- within(coa, facaa, scan = FALSE, nf = 2)

#####

#
# Plot first factorial map:
#
#####

facsp <- as.factor(rep(c("lm", "tb", "tc"),
                      c(ncol(tablm), ncol(tabtb), ncol(tabt
s.class(scua$co, fac = facsp, cstar = 0, label = "",
        col = c("green", "red", "blue"), cell=0, cpoint=0.
```

```
sub="First factorial map for synonymous codon usage"

s.label(scua$li, add.plot = TRUE, clab = 0.75)

legend( x = 0.1, y = -0.4, pch = 19, col = c("green", "red", "blue"),
        legend = c(expression(italic("Leishmania major")),
                    expression(italic("Trypanosoma brucei")),
                    expression(italic("Trypanosoma cruzi"))),
        xjust = 0, cex = 0.8)
```

```
#####
#
# END
#
#####
```

References

- [1] BENZECRI, J. P. History and prehistory of data analysis. V: The analysis of correspondence, systematic index (french), *Cahiers de l'Analyse des Donnees* **2**: 9–40, (1977)
- [2] D. CARR, R. SOMOGYI AND G. MICHAELS, *Templates for Looking at Gene Expression Clustering*, Statistical Computing and Graphics Newsletter, vol. 8, pp. 20-29, (1997).
- [3] D. R. COX AND L. BRANDWOOD, *On a discriminatory problem connected with the works of Plato*, J. Roy. Statist. Soc. Ser. B, **21** (1959), pp. 195–200.
- [4] K. FELLEBERG , N. C. HAUSER, B. BRORS, A. NEUTZNER, J. D. HOHEISEL, AND M. VINGRON, *Correspondence analysis applied to microarray data*, PNAS, (2001), pp. 10781-10786.
- [5] M. GREENACRE, *Theory and Applications of Correspondence Analysis*, Academic Press, (1984).
- [6] CHARNOMORDIC, B. AND HOLMES, S. *Correspondence Analysis for Microarrays*, Statistical Graphics and Computing Newsletter, vol.12 (2001)
- [7] CHARIF D, THIOULOUSE J, LOBRY JR, PERRIERE G. *Online*

synonymous codon usage analyses with the ade4 and seqinR packages
Bioinformatics. 2005 Feb 15;21(4):545-7.

- [8] G.K.LEDGER, *Re-counting Plato*, Oxford University Press, Oxford (1989).
- [9] K. MARDIA, J. KENT, AND J. BIBBY, *Multivariate Analysis* Academic Press, (1979).
- [10] R. D. SNEE, *Graphical display of two-way contingency tables*, The American Statistician, 28 (1974), pp. 9–12.
- [11] V.R. IYER, M. EISEN, T. ROSS, T. DOUGLAS, G. SCHULER, T. MOORE, J. LEE, J. TRENT, L. STAUDT, J. HUDSON JR., M. BOGUSKI, D. LASHKARI, D. SHALON, D. BOTSTEIN AND P. BROWN., *The Transcriptional Program in the Response of Human Fibroblasts to Serum*, *Science*, (1999) 283: 83-87.
- [12] W. WONG AND Y. CUI, *Gifi Array Analyzer* analyzing microarray data with Homogeneity Analysis, <http://biowww.dfci.harvard.edu/~ycui/Gifi.htm>
<http://biowww.dfci.harvard.edu/~ycui/Gifi.html>