Modelos Matemáticos e Aplicações
Introduction to Multivariate Statistics

Jorge Cadima

Secção de Matemática (DCEB) - Instituto Superior de Agronomia (UL)

2021-22

Programme

- Matrix Theory concepts
- Principal Component Analysis (PCA)
- Linear Discriminant Analysis (LDA)
- Cluster Analysis (Prof. Pedro Silva)

Bibliography

- Johnson, R.A. & Wichern, D.W. (2007) *Applied Multivariate Statistical Analysis*, 6th ed., Pearson Prentice Hall.
- Jolliffe, I.T. (2002) *Principal Component Analysis*, 2d. ed., Springer (Springer Series in Statistics)
- Krzanowski, W.J. (1998) *Principles of Multivariate Analysis: A User's Perspective*, Oxford Science Publications.
- Morrison, D.F. (1990) *Multivariate Statistical Methods*, 3rd.ed., McGraw-Hill.

Multivariate in R:

- Everitt, B. & Hothorn, T. (2011) *An Introduction to Applied Multivariate Analysis with R*. Springer (Use R! Series).
- Zelterman, D. (2015). *Applied Multivariate Statistics with R*. Springer (Statistics for Biology and Health Series).

Raw material for PCA and LDA

Data matrix $\mathbf{X}_{n\times p}$, with sets of observations:

- on p numerical variables (columns);
- for n individuals, or experimental units (rows).

Note: Unlike in modelling, here all variables are on an equal footing.

We will consider a descriptive (geometric) approach, both to PCA and to LDA, although in both methods probabilistic/inferential notions and approaches may be used.

Goal in PCA and LDA

We seek new variables, defined from the *p* observed variables which highlight:

- in PCA: the variability between individuals;
- in LDA: the separation between known subgroups of individuals.

In both cases, the new variables are linear combinations of the *p* observed variables.

A motivation of PCA

In the traditional representation, the data matrix $\mathbf{X}_{n\times p}$ corresponds to a scatterplot of n points in $\mathbb{R}^p$:

$$\begin{array}{ccc} p \text{ axes} & \longleftrightarrow & p \text{ variables} \\ n \text{ points} & \longleftrightarrow & n \text{ individuals} \end{array}$$

This scatterplot cannot be visualised for $p > 3$.

PCA can be seen as an "optimal" dimensionality reduction technique: we seek subspaces of dimension $k < p$ where the orthogonal projection of the scatterplot preserves a maximum of variability (equivalently, looses the least variability).

With a reduction to $k=2$ or $k=3$ dimensions, we have a visualisable approximation of the scatterplot.

An example: Somers' crayfish data

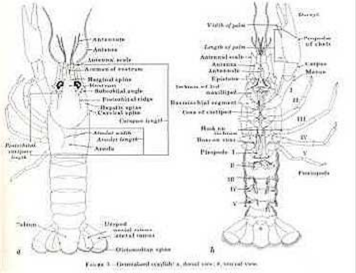
Data: $p = 13$ morphometric variables with $n = 63$ crayfish

```
> lavagantes

      x1    x2    x3    x4    x5    x6   x7   x8   x9   x10   x11   x12   x13
1  29.42 21.43 14.91 12.58 12.85 10.57 1.76 6.45 6.67  9.14 24.54 10.38 15.37
2  30.06 22.05 14.81 12.54 12.96 10.75 1.73 6.11 7.04  8.76 26.21 11.00 11.92
3  30.30 21.95 15.10 12.97 13.05 11.11 2.05 6.46 7.14  9.35 26.55 11.84 16.50
4  30.75 21.91 15.89 12.85 13.75 10.75 1.71 6.62 6.84  9.53 25.35 11.60 15.47
5  31.06 20.37 15.83 13.15 13.37 11.50 2.15 5.96 7.09  9.15 26.88 11.92 17.24
6  31.27 24.04 17.45 14.49 14.77 12.64 2.06 6.59 7.43 10.75 31.60 14.32 18.95
7  31.39 21.91 15.96 13.41 13.74 11.79 2.03 6.40 6.89  9.82 28.16 12.53 16.90
8  31.51 23.63 15.95 13.14 13.89 11.74 1.94 6.26 6.81  9.36 26.09 11.15 15.48
9  32.12 22.81 16.06 13.29 13.80 12.14 2.02 6.47 7.00  9.70 27.01 11.22 16.65
10 32.40 22.96 16.69 13.82 14.30 12.06 2.03 6.14 7.27  9.53 29.34 12.59 17.90

..............................................................................

56 33.44 24.72 17.06 14.25 16.74 12.42 2.04 6.52 7.25 10.21 26.92 11.40 16.23
57 33.48 25.32 17.50 14.15 17.20 12.40 2.17 6.94 7.54 10.37 26.85 11.40 16.34
58 33.57 25.00 16.74 14.10 16.49 12.43 1.95 7.27 7.37 10.15 25.13 11.23 14.98
59 33.74 25.30 17.11 14.26 16.35 12.37 2.26 6.82 7.41 11.14 26.43 10.91 16.02
60 34.37 25.35 17.98 14.49 16.95 12.69 2.02 7.04 7.35 10.33 27.97 11.75 17.19
61 34.66 25.32 18.50 14.16 17.37 12.60 2.32 6.88 7.59 11.00 27.76 11.87 17.58
62 34.93 26.77 18.00 14.13 16.89 12.67 2.04 7.14 7.79 10.36 26.98 11.55 17.20
63 35.73 25.79 18.35 15.06 17.15 13.14 2.15 7.09 7.83 10.59 28.29 12.30 17.45
```

These $13 \times 63 = 819$ values define a 63-point scatterplot in $\mathbb{R}^{13}$.

Somers' crayfish (cont.)

Dataset `lavagantes`: full variable names



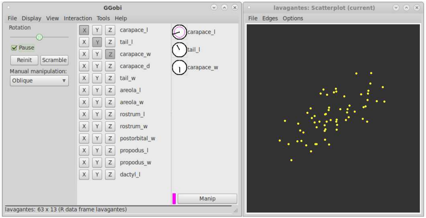
Thoma, Roger F., *A Field Guide to the Crayfishes of Obed Wild and Scenic River*, `www.nps.gov`.

x1	carapace_l	carapace length
x2	tail_l	tail length
x3	carapace_w	carapace width
x4	carapace_d	carapace depth
x5	tail_w	tail width
x6	areola_l	areola length
x7	areola_w	areola width
x8	rostrum_l	rostrum length
x9	rostrum_w	rostrum width
x10	postorbital_w	post-orbital width
x11	propodus_l	propodus length
x12	propodus_w	propodus width
x13	dactyl_l	dactyl length

Graphical representation of multivariate data

For $p = 3$ the usual representation of the data is possible, with the help of software such as the `rggobi` package, which accesses the software Ggobi from within R[1].

```
> library(rggobi)
> ggobi(lavagantes)
```



Manual manipulation: Oblique

lavagantes: 63 x 13 (R data frame lavagantes)

But for $p > 3$ we continue to have only partial visions, resulting from orthogonal projections of the $n = 63$ point scatterplot in $\mathbb{R}^{13}$ onto 3-dimensional spaces.

[1] `GGobi` is a separate, free and open source software (`www.ggobi.org`)

Orthogonal projections

Any projection impoverishes the representation: only partial visions are provided. Distances are distorted.

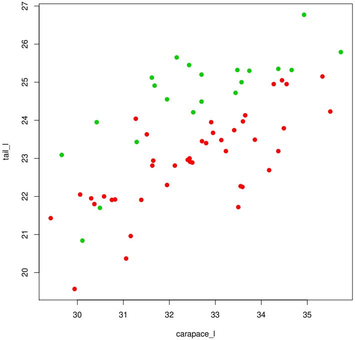
But,

- Why project only onto coordinate (hyper)planes (defined by the variable axes)? Why not other (hyper)planes?
- What is the (hyper)plane where the projection is most faithful?
- What does a “faithful projection” mean (which crietrion)?

Intuitive idea: the subspace where we project should preserve as much variability of the scatterplot as possible. This is the approach that leads to Principal Component Analysis.

Motivation: Linear Discriminant Analysis (LDA)

PCA treats all individuals on the same footing. But the first 42 crayfish are males (21 reproducing and 21 non-reproducing males) and the last 21 are females. A scatterplot of the first two variables suggests that the separation of these subgroups may be visible on the morphometric variables.



LDA: identify linear combinations of variables that best separate the subgroups.

Concepts: Types of square matrices

A matrix is square if it has the same number of rows and columns.

Here are some important types of square matrices, $\mathbf{A}_{p\times p}$:

A Diagonal	$a_{ij}=0$ if $i\neq j$ (if there is an i such that $a_{ii}\neq 0$)
$\mathbf{I}_p$ Identity	$\mathbf{A}=\mathbf{I}_p \iff a_{ij}=\begin{cases} 0 & \text{for } i\neq j \\ 1 & \text{for } i=j \end{cases}$
$\mathbf{A}^{-1}$ Inverse of **A**	$\mathbf{A}^{-1}\mathbf{A}=\mathbf{A}\mathbf{A}^{-1}=\mathbf{I}_p$ (may not exist, if it exists it is unique)
A Symmetric	$\mathbf{A}^t=\mathbf{A} \iff a_{ij}=a_{ji}, \quad \forall i,j$
A Idempotent	$\mathbf{A}^2=\mathbf{A}\mathbf{A}=\mathbf{A}$
A Orthogonal	$\mathbf{A}^{-1}=\mathbf{A}^t \iff \mathbf{A}^t\mathbf{A}=\mathbf{A}\mathbf{A}^t=\mathbf{I}_p$

Both the columns and the rows of an orthogonal matrix are orthonormal sets of vectors: vectors $\vec{\mathbf{a}}_i$ with norm one ($\|\vec{\mathbf{a}}_i\|=\sqrt{\vec{\mathbf{a}}_i^t\vec{\mathbf{a}}_i}=1$) and mutually orthogonal ($\vec{\mathbf{a}}_i^t\vec{\mathbf{a}}_j=0$, if $i\neq j$).

Matrices of (co-)variances

Symmetric matrices are important in Statistics: (co)variance and correlation matrices are symmetric matrices.

(Co-)variance matrices of $n \times p$ datasets are of the form

$$\mathbf{S} = \tfrac{1}{n-1}\,\mathbf{X}^{\mathrm{c}\,t}\mathbf{X}^{\mathrm{c}}\ ,$$

where $\mathbf{X}^{\mathrm{c}}$ is the $n \times p$ matrix whose columns are the p centred vectors of observations, $\vec{\mathbf{x}}_j^c$, i.e., the matrix with generic element $x_{ij} - \overline{x}_{.j}$:

$$\mathbf{s}_{jk} \ = \ \frac{1}{n-1}\left[\mathbf{X}^{\mathrm{c}\,t}\mathbf{X}^{\mathrm{c}}\right]_{jk} \ = \ \frac{1}{n-1}\sum_{i=1}^{n}(x_{ij} - \overline{x}_{.j})(x_{ik} - \overline{x}_{.k}) \ = \ Cov_{\vec{\mathbf{x}}_j,\vec{\mathbf{x}}_k}$$

The eigenvalues of (co-)variance matrices are always non-negative and, if there is no multicollinearity of the centred variables, they are positive.

Eigenvalues/vectors (Valores e vectores próprios)

Definition: Eigenvalues/eigenvectors

Given a square real matrix $\mathbf{A}_{p\text{x}p}$, a non-zero vector $\vec{\mathbf{x}} \in \mathbb{C}^p$ is called an eigenvector of $\mathbf{A}$, and $\lambda \in \mathbb{C}$ is its eigenvalue, if:

$$\mathbf{A}\vec{\mathbf{x}} = \lambda\vec{\mathbf{x}} \ .$$

Eigenvalues and eigenvectors of symmetric matrices

If $\mathbf{A}_{p\times p}$ is a symmetric matrix, its eigenvalues/vectors have good properties:

- Its eigenvalues and eigenvectors are always real.
- Eigenvectors corresponding to different eigenvalues are orthogonal to each other.
- Even if there are repeated eigenvalues, it is possible to determine an orthonormal set of p eigenvectors (and p corresponding eigenvalues).

The Spectral Decomposition of a symmetric matrix

Spectral Decomposition Theorem

Let $\mathbf{A}_{p\times p}$ be a symmetric matrix. Let:

- $\{\vec{\mathbf{v}}_i\}_{i=1}^{p}$ be an orthonormal set of eigenvectors; and
- $\{\lambda_i\}_{i=1}^{p}$ their corresponding p eigenvalues.

Define:

- the diagonal matrix $\mathbf{\Lambda}_{p\times p}$ whose diagonal elements are λ_i; and
- a (necessarily orthogonal) matrix $\mathbf{V}_{p\times p}$, with columns $\vec{\mathbf{v}}_i$;

Then:

$$\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^t \qquad \Longleftrightarrow \qquad \mathbf{A} = \sum_{i=1}^{p} \lambda_i \vec{\mathbf{v}}_i \vec{\mathbf{v}}_i^t \;.$$

The eigenvalues and eigenvectors are the essence of a symmetric matrix **A**.

Remarks about spectral decompositions

$$\mathbf{A} \;=\; \mathbf{V}\mathbf{\Lambda}\mathbf{V}^t \qquad \Longleftrightarrow \qquad \mathbf{A} \;=\; \sum_{i=1}^{p} \lambda_i \vec{\mathbf{v}}_i \vec{\mathbf{v}}_i^t \;.$$

- If all eigenvalues are different, the eigenvectors $\vec{\mathbf{v}}_i$ are unique, except for sign-switching (both $\vec{\mathbf{v}}_i$ and $-\vec{\mathbf{v}}_i$ are eigenvalues).
- Ordering the diagonal elements of $\mathbf{\Lambda}$ ($\lambda_1 > \lambda_2 > \ldots > \lambda_p$), the decomposition $\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^t$ is unique (except for a change of sign in any column of $\mathbf{V}$).
- If there are equal eigenvalues, the decomposition is not unique.

  In fact, let $\vec{\mathbf{x}}_1$ and $\vec{\mathbf{x}}_2$ be eigenvectors of $\mathbf{A}$ sharing a common eigenvalue λ. Since $\mathbf{A}\vec{\mathbf{x}}_1 = \lambda\vec{\mathbf{x}}_1$ e $\mathbf{A}\vec{\mathbf{x}}_2 = \lambda\vec{\mathbf{x}}_2$, we have:

$$\mathbf{A}(\alpha\vec{\mathbf{x}}_1 + \beta\vec{\mathbf{x}}_2) \;=\; \alpha\mathbf{A}\vec{\mathbf{x}}_1 + \beta\mathbf{A}\vec{\mathbf{x}}_2 \;=\; \alpha\cdot\lambda\vec{\mathbf{x}}_1 + \beta\cdot\lambda\vec{\mathbf{x}}_2 \;=\; \lambda\left(\alpha\vec{\mathbf{x}}_1 + \beta\vec{\mathbf{x}}_2\right) \;,$$

  hence $\alpha\vec{\mathbf{x}}_1 + \beta\vec{\mathbf{x}}_2$ is also an eigenvector of $\mathbf{A}$, with the same eigenvalue λ. All vectors of the subspace spanned by $\vec{\mathbf{x}}_1$ and $\vec{\mathbf{x}}_2$ are eigenvectors with the same eigenvalue λ.

Traces (of square matrices)

Let **A** be a square matrix:

- The trace of **A** is defined as the sum of its diagonal elements:

$$\mathrm{tr}(\mathbf{A}) = \sum_{i=1}^{p} a_{ii}.$$

- The trace is a linear operator, that is,

$$\mathrm{tr}(\alpha\,\mathbf{A} + \beta\,\mathbf{B}) \quad = \quad \alpha\,\mathrm{tr}(\mathbf{A}) + \beta\,\mathrm{tr}(\mathbf{B})$$

The inner product of two matrices of the same size, $\mathbf{A}_{n\times p}$ and $\mathbf{B}_{n\times p}$, is usually defined as:

$$<\mathbf{A},\mathbf{B}> = \ \mathrm{tr}(\mathbf{A}^t\mathbf{B}) \ = \ \sum_{j=1}^{p} (\mathbf{A}^t\mathbf{B})_{jj} \ = \ \sum_{i=1}^{n}\sum_{j=1}^{p} a_{ij} b_{ij} \ .$$

Circularity of the trace

Product of two matrices: $\mathbf{A}_{n\times p}, \mathbf{B}_{p\times n} \quad \Longrightarrow \quad \mathrm{tr}(\mathbf{AB}) = \mathrm{tr}(\mathbf{BA})$.

(Even when $\mathbf{AB} \neq \mathbf{BA}$: both traces are $\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{p} a_{ij}b_{ji}$)

Product of 3 matrices: $\mathbf{A}_{m\times k}, \mathbf{B}_{k\times p}, \mathbf{C}_{p\times m} \quad \Rightarrow \mathrm{tr}(\mathbf{ABC}) = \mathrm{tr}(\mathbf{BCA})$.
(Apply the previous result to the two matrices **A** and **BC**)

Product of n matrices: If $\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3, ..., \mathbf{A}_n$ are matrices of size $(p_0 \times p_1)$, $(p_1 \times p_2)$, $(p_2 \times p_3)$, ..., $(p_{n-1} \times p_0)$, then,

$$\mathrm{tr}(\mathbf{A}_1\mathbf{A}_2\mathbf{A}_3\cdots\mathbf{A}_n) = \mathrm{tr}(\mathbf{A}_2\mathbf{A}_3\cdots\mathbf{A}_n\mathbf{A}_1)\ .$$

(Apply the first result to matrices $\mathbf{A}_1$ and $\mathbf{A}_2\mathbf{A}_3\cdots\mathbf{A}_n$)

From the Spectral Decomposition, it is easy to see that the trace of a symmetric matrix **A** is also the sum of its eigenvalues:

$$\mathrm{tr}(\mathbf{A}) = \mathrm{tr}(\mathbf{V\Lambda V}^t) = \mathrm{tr}(\mathbf{\Lambda}\underbrace{\mathbf{V}^t\mathbf{V}}_{=\mathbf{I}_p}) = \mathrm{tr}(\mathbf{\Lambda}) = \sum_{i=1}^{p}\lambda_i\ .$$

PCA: a statistical approach

A frequent way of introducing PCA uses statistical concepts.

Given the data matrix $\mathbf{X}_{n\times p}$ (each column associated with a variable, and each row with an observed individual), we seek the linear combination of the p variables with maximum variance.

That is, we seek the vector $\vec{\mathbf{v}} = (v_1, v_2, ..., v_p) \in \mathbb{R}^p$ such that

$$\mathbf{X}\vec{\mathbf{v}} \;=\; v_1\,\vec{\mathbf{x}}_1 + v_2\,\vec{\mathbf{x}}_2 + v_3\,\vec{\mathbf{x}}_3 + ... + v_p\,\vec{\mathbf{x}}_p$$

has maximum variance (with $\vec{\mathbf{x}}_j \in \mathbb{R}^n$ the vector of observations of variable j, i.e., the j-th column of $\mathbf{X}$).

The variance of $\mathbf{X}\vec{\mathbf{v}}$ is given by $\vec{\mathbf{v}}^t\mathbf{S}\vec{\mathbf{v}}$, where $\mathbf{S}$ is the dataset's (co)variance matrix. Thus, we seek the vector $\vec{\mathbf{v}}$ that maximises $\vec{\mathbf{v}}^t\mathbf{S}\vec{\mathbf{v}}$.

Variance of linear combinations of variables

Let $\mathbf{S}$ be the matrix of (co)variances defined by a data matrix $\mathbf{X}$.

The variance of a linear combination of the columns of $\mathbf{X}$, $\vec{\mathbf{y}} = \mathbf{X}\vec{\mathbf{a}}$, is the quadratic form of $\mathbf{S}$ defined by $\vec{\mathbf{a}}$:

$$var(\vec{\mathbf{y}}) \ = \ var(\mathbf{X}\vec{\mathbf{a}}) \ = \ \vec{\mathbf{a}}^t\mathbf{S}\vec{\mathbf{a}} \ .$$

In fact, $\vec{\mathbf{a}}^t\mathbf{S}\vec{\mathbf{a}} \ = \ \frac{1}{n-1}\,\vec{\mathbf{a}}^t\mathbf{X}^{c\,t}\mathbf{X}^c\vec{\mathbf{a}} \ = \ \frac{1}{n-1}\,\|\mathbf{X}^c\vec{\mathbf{a}}\|^2$, and

$$\mathbf{X}^c\vec{\mathbf{a}} \ = \ (\mathbf{I}_n - \mathbf{P}_{\vec{\mathbf{1}}_n})\mathbf{X}\vec{\mathbf{a}} \ = \ \mathbf{X}\vec{\mathbf{a}} - \mathbf{P}_{\vec{\mathbf{1}}_n}\mathbf{X}\vec{\mathbf{a}} \ = \ \vec{\mathbf{y}} - (\overline{y})\vec{\mathbf{1}}_n$$

is the centred vector for the linear combination $\vec{\mathbf{y}} = \mathbf{X}\vec{\mathbf{a}}$, with generic element $y_i^c = y_i - \overline{y}$.
Thus, $\vec{\mathbf{a}}^t\mathbf{S}\vec{\mathbf{a}}$ is the sample variance of $\vec{\mathbf{y}} = \mathbf{X}\vec{\mathbf{a}}$:

$$\vec{\mathbf{a}}^t\mathbf{S}\vec{\mathbf{a}} \ = \ \frac{1}{n-1}\,\|\mathbf{X}^c\vec{\mathbf{a}}\|^2 \ = \ \frac{1}{n-1}\sum_{i=1}^{n}(y_i - \overline{y})^2 \ = \ var(\vec{\mathbf{y}}) \ .$$

The covariance between different linear combinations, $\mathbf{X}\vec{\mathbf{a}}$ and $\mathbf{X}\vec{\mathbf{b}}$, is:

$$Cov[\mathbf{X}\vec{\mathbf{a}}, \mathbf{X}\vec{\mathbf{b}}] = \vec{\mathbf{a}}^t\mathbf{S}\vec{\mathbf{b}} \ .$$

Statistical approach (cont.)

Without additional restrictions, the problem of maximising $\vec{\mathbf{v}}^t\mathbf{S}\vec{\mathbf{v}}$ cannot be solved: we could choose arbitrarily large elements in vector $\vec{\mathbf{v}}$.

Impose the restriction of considering only unit-norm vectors (sum of squared vector coefficients equal to 1), that is, vectors of the form $\frac{\vec{\mathbf{v}}}{\|\vec{\mathbf{v}}\|}$ (with $\vec{\mathbf{v}} \neq \vec{\mathbf{0}}$).

Hence, the problem is to maximise the so-called Rayleigh-Ritz ratio of $\mathbf{S}$:

$$\max_{\vec{\mathbf{v}}\in\mathbb{R}^p\setminus\{\vec{\mathbf{0}}\}} \left[\frac{\vec{\mathbf{v}}}{\|\vec{\mathbf{v}}\|}\right]^t \mathbf{S}\,\frac{\vec{\mathbf{v}}}{\|\vec{\mathbf{v}}\|} = \max_{\vec{\mathbf{v}}\in\mathbb{R}^p\setminus\{\vec{\mathbf{0}}\}} \frac{\vec{\mathbf{v}}^t\mathbf{S}\vec{\mathbf{v}}}{\|\vec{\mathbf{v}}\|^2} = \max_{\vec{\mathbf{v}}\in\mathbb{R}^p\setminus\{\vec{\mathbf{0}}\}} \frac{\vec{\mathbf{v}}^t\mathbf{S}\vec{\mathbf{v}}}{\vec{\mathbf{v}}^t\vec{\mathbf{v}}}$$

The solution is given by the eigenvector $\vec{\mathbf{v}}_1$ (of norm 1), associated with the largest eigenvalue of $\mathbf{S}$, λ_1.

Rayleigh-Ritz Theorem

Let $\mathbf{A}_{p\times p}$ be a symmetric matrix, with eigenvalues in decreasing order: $\lambda_{\max} = \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_{p-1} \geq \lambda_p = \lambda_{\min}$.

- The largest eigenvalue of $\mathbf{A}$ verifies: $\lambda_{\max} = \max\limits_{\vec{\mathbf{x}}\neq\vec{\mathbf{0}}} \frac{\vec{\mathbf{x}}^t\mathbf{A}\vec{\mathbf{x}}}{\vec{\mathbf{x}}^t\vec{\mathbf{x}}}$,

  when $\vec{\mathbf{x}} = \vec{\mathbf{v}}_1$, the eigenvector associated with λ_{max}.

- The smallest eigenvalue of $\mathbf{A}$ verifies: $\lambda_{\min} = \min\limits_{\vec{\mathbf{x}}\neq\vec{\mathbf{0}}} \frac{\vec{\mathbf{x}}^t\mathbf{A}\vec{\mathbf{x}}}{\vec{\mathbf{x}}^t\vec{\mathbf{x}}}$,

  when $\vec{\mathbf{x}} = \vec{\mathbf{v}}_p$, the eigenvector associated with λ_{min}.

- The remaining eigenvalues (λ_i)/eigenvectors$(\vec{\mathbf{v}}_i)$ of $\mathbf{A}$ which can also be characterised from the Rayleigh-Ritz ratio of $\mathbf{A}$:

$$\lambda_j = \max_{(\vec{\mathbf{x}}\perp\vec{\mathbf{v}}_1,\vec{\mathbf{v}}_2,\ldots\vec{\mathbf{v}}_{j-1})\wedge(\vec{\mathbf{x}}\neq\vec{\mathbf{0}})} \frac{\vec{\mathbf{x}}^t\mathbf{A}\vec{\mathbf{x}}}{\vec{\mathbf{x}}^t\vec{\mathbf{x}}}$$

$$\lambda_j = \min_{(\vec{\mathbf{x}}\perp\vec{\mathbf{v}}_{j+1},\vec{\mathbf{v}}_{j+2},\ldots\vec{\mathbf{v}}_{p})\wedge(\vec{\mathbf{x}}\neq\vec{\mathbf{0}})} \frac{\vec{\mathbf{x}}^t\mathbf{A}\vec{\mathbf{x}}}{\vec{\mathbf{x}}^t\vec{\mathbf{x}}}$$

  with the equalities associated with $\vec{\mathbf{x}} = \vec{\mathbf{v}}_j$.

The first Principal Component

The first Principal Component is the linear combination $\vec{\mathbf{y}}_1 = \mathbf{X}\vec{\mathbf{v}}_1$, with $\vec{\mathbf{v}}_1$ the eigenvector associated with the largest eigenvalue of $\mathbf{S}$.

Note: If $\vec{\mathbf{v}}$ is an eigenvector, so is $-\vec{\mathbf{v}}$. The solutions define straight lines, but do not define specific directions on those lines. Just as the eigenvector $\vec{\mathbf{v}}_1$, so too the first PC, $\mathbf{X}\vec{\mathbf{v}}_1$, is defined up to a multiplication by -1.

The vector of coefficients $\vec{\mathbf{v}}_1$ defines the line in $\mathbb{R}^p$ of maximum variance for the *n*-point scatterplot defined by the data.

Eigenvalue λ_1 is the variance of the first PC:

$$var(\vec{\mathbf{y}}_1) = var(\mathbf{X}\vec{\mathbf{v}}_1) = \vec{\mathbf{v}}_1^t\mathbf{S}\vec{\mathbf{v}}_1 = \vec{\mathbf{v}}_1 \cdot \lambda_1\vec{\mathbf{v}}_1 = \lambda_1 \cdot \vec{\mathbf{v}}_1^t\vec{\mathbf{v}}_1 = \lambda_1 \ .$$

The larger λ_1, the more elongated is the $\mathbb{R}^p$ scatterplot in the direction defined by the first PC.

PCA: statistical approach (cont.)

Having defined the first PC, we seek a new linear combination $\vec{\mathbf{y}} = \mathbf{X}\vec{\mathbf{v}}$ (with $\vec{\mathbf{v}}^t\vec{\mathbf{v}} = 1$) of maximum variance, uncorrelated with PC 1.

Zero correlation means zero covariance. The covariance of two linear combinations of the columns of matrix $\mathbf{X}$, $\mathbf{X}\vec{\mathbf{v}}_1$ and $\mathbf{X}\vec{\mathbf{v}}$, is given by $\vec{\mathbf{v}}^t\mathbf{S}\vec{\mathbf{v}}_1$, where $\mathbf{S}$ is the covariance matrix for the data in $\mathbf{X}$.

But $\vec{\mathbf{v}}_1$ is an eigenvector of $\mathbf{S}$, with eigenvalue λ_1. Hence:

$$\mathrm{cov}\left(\mathbf{X}\vec{\mathbf{v}}, \mathbf{X}\vec{\mathbf{v}}_1\right) = \vec{\mathbf{v}}^t\mathbf{S}\vec{\mathbf{v}}_1 = 0 \quad \Leftrightarrow \quad \lambda_1\vec{\mathbf{v}}^t\vec{\mathbf{v}}_1 = 0 \quad \Leftrightarrow \quad \vec{\mathbf{v}} \perp \vec{\mathbf{v}}_1 \ .$$

Thus, maximising the variance of $\mathbf{X}\vec{\mathbf{v}}$, given uncorrelatedness of $\mathbf{X}\vec{\mathbf{v}}$ with $\mathbf{X}\vec{\mathbf{v}}_1$ is equivalent to maximising $\frac{\vec{\mathbf{v}}^t\mathbf{S}\vec{\mathbf{v}}}{\vec{\mathbf{v}}^t\vec{\mathbf{v}}}$, subject to $\vec{\mathbf{v}}$ being orthogonal with $\vec{\mathbf{v}}_1$.

The problem is again associated with Rayleigh-Ritz ratios.

PCA: statistical approach (cont.)

Maximising the variance of $\mathbf{X}\vec{\mathbf{v}}$ subject to uncorrelatedness of $\mathbf{X}\vec{\mathbf{v}}$ and $\mathbf{X}\vec{\mathbf{v}}_1$ means taking $\vec{\mathbf{v}} = \pm\vec{\mathbf{v}}_2$, the eigenvector of $\mathbf{S}$ associated with its second largest eigenvalue, λ_2.

$\vec{\mathbf{y}}_2 = \pm\mathbf{X}\vec{\mathbf{v}}_2$ is the second principal component, with variance λ_2.

PCs are solutions to the problem of finding successive uncorrelated linear combinations of maximum variance.

The j-th principal component is given by $\vec{\mathbf{y}}_j = \pm\mathbf{X}\vec{\mathbf{v}}_j$, where $\vec{\mathbf{v}}_j$ is the eigenvector of $\mathbf{S}$ associated with the j-th largest eigenvalue λ_j.

The variance of the j-th PC is given by the corresponding eigenvalue: $var(\vec{\mathbf{y}}_j) = \lambda_j$.

PCA in R

The usual command to perform a PCA in `R` is the command `prcomp`.

Command `prcomp` has a single compulsory argument: the name of the `data.frame` or `matrix` with the data (each column corresponding to a variable).

As with other `R` commands, the result is an object of class `list`, containing different information regarding the results of the analysis.

Note: There is an alternative `princomp` command. But for various reasons, including numerical accuracy in the case of nearly singular (almost non-invertible) covariance matrices the command `prcomp` is preferable.

The command prcomp

PCA - Crayfish data

```
> lav.acp <- prcomp(lavagantes)
> lav.acp

Standard deviations (1, .., p=13):
[1] 4.4171243 2.1583124 0.9617894 0.7071970 0.6163559 0.4992560 0.4639879  <- standard deviation
[8] 0.3848417 0.3362918 0.2500701 0.2060563 0.1770375 0.1405790                  of each PC

Rotation (n x k) = (13 x 13):
                     PC1         PC2         PC3         PC4          PC5
carapace_l    0.28762060  0.36935786  0.08475822 -0.31404094 -0.454639049   <- each column is an
tail_l        0.10615292  0.61487598 -0.01728674  0.46421995  0.550775374      eigenvector v_j
carapace_w    0.19089393  0.22112280  0.09978650 -0.10987953 -0.186701149      of the data's
carapace_d    0.13951311  0.14784642  0.13138041  0.01598041  0.105009202      (co)variance
tail_w        0.04682070  0.49290700 -0.05172379  0.06592005 -0.405755003      matrix. These
areola_l      0.13858508  0.15588574 -0.03136931 -0.78849399  0.514893584      vectors contain the
areola_w      0.02862658  0.02088959 -0.05104427 -0.01123927 -0.005062728      coefficients of the
rostrum_l     0.04321132  0.10238463 -0.00534869  0.10538116 -0.015312405      linear combinations
rostrum_w     0.06381638  0.06445436  0.05636521 -0.02008425 -0.071806372      defining the PCs.
postorbital_w 0.08947075  0.12850014  0.07576734 -0.01777992  0.021872310
propodus_l    0.70705994 -0.28621233  0.04885310  0.16407517  0.077728529
propodus_w    0.31334632 -0.14849063  0.69820134  0.07580938  0.026674997
dactyl_l      0.46456390 -0.10926197 -0.67839228  0.05350023 -0.040805783
[...]           <- the remaining vectors of coefficients were omitted, for reasons of space.
```

The coefficients of each linear combination (columns of the `Rotation` object) are called the PC loadings.

Properties of PCs

- The sum of variances (inertia) of the p principal components is equal to the sum of variances of the p original variables:

$$\sum_{i=1}^{p} s_i^2 = \mathrm{tr}(\mathbf{S}) = \mathrm{tr}(\mathbf{V\Lambda V}^t) = \mathrm{tr}(\mathbf{\Lambda V}^t\mathbf{V}) = \mathrm{tr}(\mathbf{\Lambda}) = \sum_{i=1}^{p} \lambda_i \ .$$

- Thus, we can say that the j-th PC accounts for a proportion of the total variability (inertia) equal to $\pi_j \ = \ \frac{\lambda_j}{\sum_{i=1}^{p} \lambda_i}$.
- This measure can be extended to subsets of principal components. The first q PCs account for

$$\sum_{i=1}^{q} \pi_i \times \ 100\% = \frac{\sum_{i=1}^{q} \lambda_i}{\sum_{j=1}^{p} \lambda_j} \ \times \ 100\%$$

of the total variability (inertia) of the dataset.

The command `summary`

PCA - Crayfish data

> summary(lav.acp)

```
Importance of components:
           PC1   PC2    PC3    PC4    PC5     PC6     PC7     PC8     PC9    PC10    PC11    PC12    PC13
Std. Dev. 4.417 2.158 0.9618 0.7072 0.6164 0.49926 0.46399 0.38484 0.33629 0.25007 0.20606 0.17704 0.14058
Prop.Var. 0.727 0.173 0.0344 0.0186 0.0141 0.00928 0.00802 0.00551 0.00421 0.00233 0.00158 0.00117 0.00074
Cum.Prop. 0.727 0.900 0.9344 0.9530 0.9672 0.97645 0.98446 0.98998 0.99419 0.99652 0.99810 0.99926 1.00000
```

On the line associated with the first principal component we preserve 72.7% of the dataset's total variability.

On the plane associated with the first two principal components we preserve 90.0% of the dataset's total variability.

The three-dimensional subspace defined by the first three PCs preserves 93.4% of the total variability.

With a 3-dimensional representation, only some 6.6% of the total variability is not visualised.

Vectors of scores

By default, prcomp does not show the scores of each individual on a given PC, i.e., the value of each individual on the linear combination $\vec{\mathbf{y}}_j = \mathbf{X}\vec{\mathbf{v}}_j$.

The scores are stored in the list created when invoking the `prcomp` command, in an object called `x`:

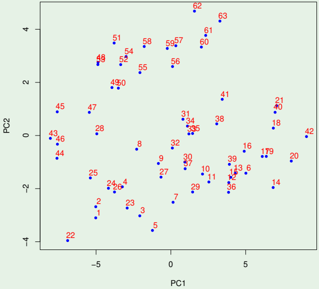
```
> names(lav.acp)

[1] "sdev"     "rotation" "center"   "scale"    "x"

> lav.acp$x

          PC1        PC2         PC3          PC4         PC5          PC6
1  -5.0216041 -3.09975004 -0.93638716  0.590170762  0.34242883 -0.311295721
2  -5.0199046 -2.68138921  1.93090666  0.652936303  0.71306147  2.411219117
3  -2.0772687 -3.02373521 -0.44934354  0.613510708  0.54941375 -0.365822245
[...]
62  1.5767872  4.68339718 -0.49231884  0.246787192 -0.11313707  0.138658304
63  3.2782407  4.30830749  0.15373020 -0.562657698 -0.73379507  0.200035217
[...]
```

These are the coordinates used in the low-dimensional scatterplots that best preserve the dataset's variability.

The best 2-dimensional representation

First principal plane for the crayfish data

```
> plot(lav.acp$x[,1:2],col="blue", pch=16, cex=0.8)
> text(lav.acp$x[,1:2]+0.2, label=rownames(lavagantes), col="red")
```

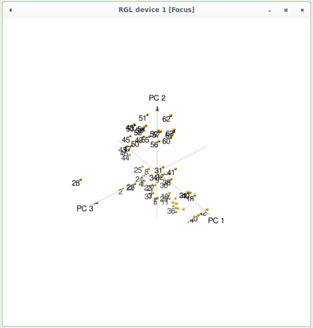


Individuals 43 to 63 are females, the others males. PCA did not use that information, but is reflecting its effect on the morphometric characteristics.

The best 3-dimensional representation

Package pca3d creates and enables us to rotate a 3-D scatterplot defined by the scores for the first 3 PCs:

```
> library(pca3d)
> pca3d(lav.acp, show.labels=TRUE)
```

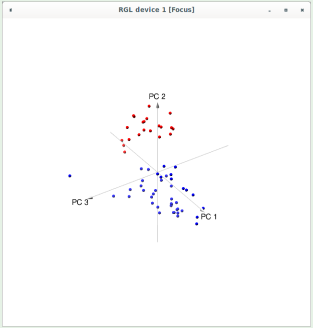


We can see that the third PC separates an outlying observation 28 from the others. So outliers may be identifiable on some PCs.

The best 3-D representation (cont.)

Package pca3d allows us to use different colours for individuals:

```
> pca3d(lav.acp, col=rep(c("blue","red"),c(42,21)))
```

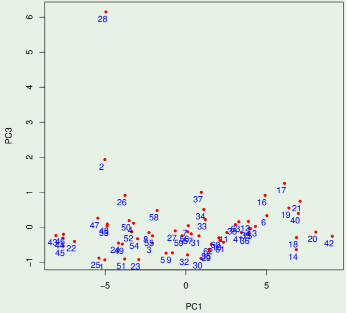


The fact that the first two PCs separate males (blue) and females (red) is highlighted.

Individual 28 and the third PC

Outlier in the crayfish data

```
> plot(lav.acp$x[,c(1,3)],col="red", pch=16, cex=0.8)
> text(lav.acp$x[,c(1,3)]-0.2, label=rownames(lavagantes), col="blue")
```

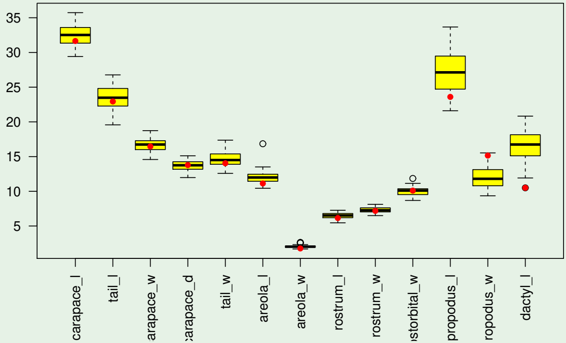


Individual 28 contributes heavily towards the third orthogonal direction of maximum variability. Why? What is different in individual 28?

Revisiting individual 28

Outlier in the crayfish data

```
> boxplot(lavagantes, col="yellow",las=2)
> points(1:13,lavagantes[28,], pch=16, col="red")
```



Individual 28 has unusual measurements in its claws.

Eigenvalue decomposition

The information produced by the command `prcomp` could be obtained with the spectral decomposition of the dataset's covariance matrix, using the command `eigen`:

```
> eigen(var(lavagantes))

$values                    <-- eigenvalues
[1]  19.51098705  4.65831240  0.92503887  0.50012760  0.37989465  0.24925657
[7]   0.21528474  0.14810313  0.11309220  0.06253506  0.04245919  0.03134228
[13]  0.01976246

$vectors                   <-- eigenvectors
             [,1]        [,2]        [,3]        [,4]         [,5]         [,6]
[1,]  -0.28762060 -0.36935786 -0.08475822  0.31404094 -0.454639049  0.272071976
[2,]  -0.10615292 -0.61487598  0.01728674 -0.46421995  0.550775374  0.088028646
[3,]  -0.19089393 -0.22112280 -0.09978650  0.10987953 -0.186701149 -0.178125878
[4,]  -0.13951311 -0.14784642 -0.13138041 -0.01598041  0.105009202 -0.171612241
[5,]  -0.04682070 -0.49290700  0.05172379 -0.06592005 -0.405755003 -0.046182873
[6,]  -0.13858508 -0.15588574  0.03136931  0.78849399  0.514893584 -0.004876079
[7,]  -0.02862658 -0.02088959  0.05104427  0.01123927 -0.005062728  0.026873555
[8,]  -0.04321132 -0.10238463  0.00534869 -0.10538116 -0.015312405 -0.029408152
[9,]  -0.06381638 -0.06445436 -0.05636521  0.02008425 -0.071806372  0.007891374
[10,] -0.08947075 -0.12850014 -0.07576734  0.01777992  0.021872310 -0.276900583
[11,] -0.70705994  0.28621233 -0.04885310 -0.16407517  0.077728529  0.541197594
[12,] -0.31334632  0.14849063 -0.69820134 -0.07580938  0.026674997 -0.476061633
[13,] -0.46456390  0.10926197  0.67839228 -0.05350023 -0.040805783 -0.506989966
[...]
```

Eigenvalue decomposition (cont.)

```
> sqrt(eigen(var(lavagantes))$val)

[1] 4.4171243 2.1583124 0.9617894 0.7071970 0.6163559 0.4992560 0.4639879
[8] 0.3848417 0.3362918 0.2500701 0.2060563 0.1770375 0.1405790
```

```
> lav.acp$sdev

[1] 4.4171243 2.1583124 0.9617894 0.7071970 0.6163559 0.4992560 0.4639879
[8] 0.3848417 0.3362918 0.2500701 0.2060563 0.1770375 0.1405790
```

```
> eigen(var(lavagantes))$vec

            [,1]        [,2]        [,3]        [,4]         [,5]         [,6]
[1,] -0.28762060 -0.36935786 -0.08475822 -0.31404094 -0.454639049 -0.272071976
[2,] -0.10615292 -0.61487598  0.01728674  0.46421995  0.550775374 -0.088028646
[3,] -0.19089393 -0.22112280 -0.09978650 -0.10987953 -0.186701149  0.178125878
[...]
```

```
> lav.acp$rot

                     PC1         PC2         PC3         PC4          PC5
carapace_l    0.28762060  0.36935786  0.08475822 -0.31404094 -0.454639049
tail_l        0.10615292  0.61487598 -0.01728674  0.46421995  0.550775374
carapace_w    0.19089393  0.22112280  0.09978650 -0.10987953 -0.186701149
[...]
```

Note: Notice how some eigenvectors differ by a factor -1.

More properties of PCs

Correlations between PCs and variables

The correlation between the i-th variable $\vec{\mathbf{x}}_i$ and the j-th PC $\mathbf{X}\vec{\mathbf{v}}_j$ is:

$$\mathrm{corr}(\vec{\mathbf{x}}_i, \mathbf{X}\vec{\mathbf{v}}_j) = \sqrt{\lambda_j} \cdot \frac{v_{ij}}{s_i}$$

s_i – standard deviation of variable $\vec{\mathbf{x}}_i$

v_{ij} – coefficient (loading) of $\vec{\mathbf{x}}_i$ in PC j

$\sqrt{\lambda_j}$ – standard deviation of the j-th PC

$\vec{\mathbf{x}}_i = \mathbf{X}\vec{\mathbf{e}}_i$, where $\vec{\mathbf{e}}_i$ is the vector whose only non-zero element is a 1 in position i (i-th vector in the canonical base for $\mathbb{R}^p$).

The covariance between the linear combinations $\mathbf{X}\vec{\mathbf{v}}_j$ and $\vec{\mathbf{x}}_i = \mathbf{X}\vec{\mathbf{e}}_i$ is $\vec{\mathbf{e}}_i^t\mathbf{S}\vec{\mathbf{v}}_j$, where $\mathbf{S}$ is the dataset's covariance matrix. Hence,

$$cor(\vec{\mathbf{x}}_i, \mathbf{X}\vec{\mathbf{v}}_j) = \frac{cov(\mathbf{X}\vec{\mathbf{e}}_i, \mathbf{X}\vec{\mathbf{v}}_j)}{\sqrt{var(\vec{\mathbf{x}}_i) \cdot var(\mathbf{X}\vec{\mathbf{v}}_j)}} = \frac{\vec{\mathbf{e}}_i^t\mathbf{S}\vec{\mathbf{v}}_j}{s_i \cdot \sqrt{\lambda_j}} = \frac{\lambda_j\vec{\mathbf{e}}_i^t\vec{\mathbf{v}}_j}{s_i \cdot \sqrt{\lambda_j}} = \sqrt{\lambda_j}\,\frac{v_{ij}}{s_i}\ .$$

Interpretation of PCs

Correlations between PCs and variables (Crayfish)

The correlations between original variables and PCs may be useful when interpreting PCs. We can use the formula above or the command:

```
> round(cor(lavagantes, lav.acp$x),d=2)
```

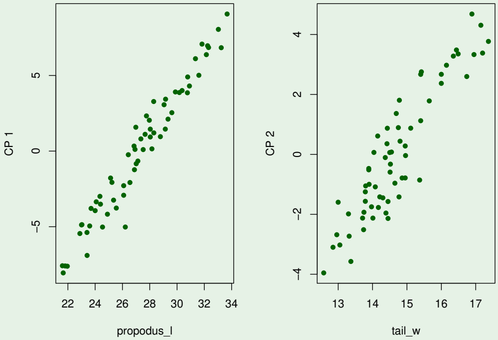
```
                PC1   PC2   PC3   PC4   PC5   PC6   PC7   PC8   PC9  PC10  PC11  PC12
carapace_l     0.81  0.51  0.05 -0.14 -0.18  0.09 -0.15  0.06 -0.03  0.02 -0.02  0.01
tail_l         0.31  0.89 -0.01  0.22  0.23  0.03 -0.04  0.06  0.02 -0.01 -0.01  0.00
carapace_w     0.83  0.47  0.09 -0.08 -0.11 -0.09  0.02 -0.02  0.09 -0.14  0.12  0.05
carapace_d     0.78  0.41  0.16  0.01  0.08 -0.11 -0.06 -0.25 -0.33 -0.03  0.03 -0.01
tail_w         0.18  0.91 -0.04  0.04 -0.21 -0.02  0.28 -0.04  0.00  0.02 -0.04 -0.01
areola_l       0.64  0.35 -0.03 -0.58  0.33  0.00  0.11  0.01  0.01  0.03  0.00  0.00
areola_w       0.60  0.21 -0.23 -0.04 -0.01  0.06  0.10  0.09 -0.03 -0.14  0.08 -0.37
rostrum_l      0.50  0.58 -0.01  0.20 -0.02 -0.04  0.03  0.03  0.01  0.49  0.35  0.04
rostrum_w      0.76  0.38  0.15 -0.04 -0.12  0.01 -0.13 -0.03  0.12 -0.02  0.12 -0.40
postorbital_w  0.65  0.45  0.12 -0.02  0.02 -0.23 -0.23 -0.40  0.29  0.08 -0.09  0.01
propodus_l     0.98 -0.19  0.01  0.04  0.01  0.08  0.03 -0.03  0.01  0.00  0.00  0.00
propodus_w     0.87 -0.20  0.42  0.03  0.01 -0.15  0.03  0.08  0.00  0.01 -0.02  0.00
dactyl_l       0.94 -0.11 -0.30  0.02 -0.01 -0.12 -0.01  0.03 -0.01  0.00 -0.01  0.00
```

PC 1 is very strongly correlated with claw measurements, in particular `propodus_l`.
PC 2 is very strongly correlated with the tail measurements, in particular `tail_w`.

Correlations between PCs and variables (cont.)

Correlations PCs/variables in crayfish data

```
> par(mfrow=c(1,2))            <- creates a ‘‘1x2 matrix of scatterplots’’
> plot(lavagantes[,11], lav.acp$x[,1], xlab="propodus_l", ylab="CP 1", pch=16, col="darkgreen")
> plot(lavagantes[,5], lav.acp$x[,2], xlab="tail_w", ylab="CP 2", pch=16, col="darkgreen")
> par(mfrow=c(1,1))            <- recreates the original graphic window
```

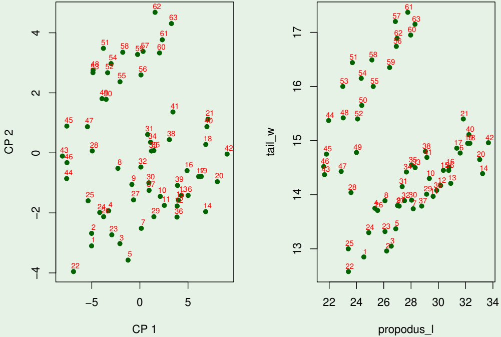


Correlations between PCs and variables (cont.)

Again the crayfish

The strong correlations suggest a scatterplot of two original variables:

```
> plot(lav.acp$x[,1:2], xlab="CP 1", ylab="CP 2", pch=16, col="darkgreen")
> text(lav.acp$x[,1:2]+0.2, label=rownames(lavagantes), col="red", cex=0.7)
> plot(lavagantes[,c(11,5)], xlab="propodus_l", ylab="tail_w", pch=16, col="darkgreen")
> text(lavagantes[,c(11,5)]+0.1, label=rownames(lavagantes), col="red", cex=0.7)
```



Correlation matrix PCA

An inconvenient characteristic of PCA is that (unlike, for example, linear regression) PCA results change if there are different changes of scale in different variables.

This sensitivity of PCA is natural, given the nature of the criterion which PCA optimizes: variance.

To overcome this problem, and since most changes of scale are linear transformations, it is common to standardise the data before carrying out a PCA:

$$x_{ij} \quad \longrightarrow \quad z_{ij} = \frac{x_{ij} - \overline{x}_{.j}}{s_j} \quad ,$$

where

- x_{ij} is the observation for individual i on variable j;
- $\overline{x}_{.j}$ is the mean of the n observations on variable j;
- s_j is the standard deviation of the n observations on variable j;
- z_{ij} is the standardised observation for individual i on variable j.

Centring, in the traditional representation in $\mathbb{R}^p$

What is the effect of centring a data matrix **X** on the scatterplot associated with the traditional representations of the data, in $\mathbb{R}^p$?

Transforming **X** into $\mathbf{X}^c$ just changes the mean of each variable, which becomes zero. Geometrically, the centre of gravity of the *n*-point scatterplot in $\mathbb{R}^p$ becomes the origin, i.e., there is a translation of the centre of gravity:

$$(\overline{x_1}, \overline{x_2}, ..., \overline{x_p}) \quad \longrightarrow \quad (0, 0, ..., 0) \ .$$

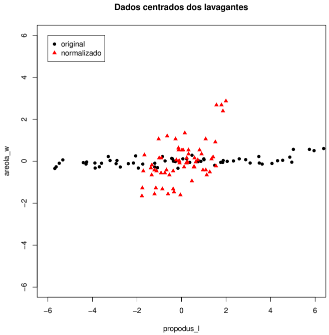
It is common to represent PCs in $\mathbb{R}^p$ with centring (that is, the scatterplot of the scores has centre of gravity at the origin).

It corresponds to considering the linear combinations of the centred variables (with the usual vectors of loadings).

Standardisation, in the representation in $\mathbb{R}^p$

What is the effect of standardising, i.e., both centring and dividing each variable by its standard deviation? All variables will now have the same variance (1). Hence, the scatterplot in $\mathbb{R}^p$ becomes more spherical.

We illustrate with the (centred) crayfish data, using only two variables: those with the largest, and the smallest, variance:



$$
\begin{aligned}
s_7^2 &= 0.04409 \\
s_{11}^2 &= 10.24217
\end{aligned}
$$

Changing the shape of the scatterplot also changes the directions of main variability.

Correlation matrix PCA (cont.)

Correlation matrices are the covariance matrices of a matrix **Z** of centred and standardised data, whose generic element is $z_{ij} = \frac{x_{ij} - \overline{x}_{\cdot j}}{s_j}$:

$$\mathbf{R} = \frac{1}{n-1} \mathbf{Z}^t \mathbf{Z} \ .$$

Thus, a PCA on standardised data is known as a Correlation Matrix PCA.

In a correlation matrix PCA,

- Principal Components are linear combinations of the standardised data;
- The loadings (coefficients) of those linear combinations are given by successive eigenvectors of the correlation matrix **R**;
- the variances of successive PCs are given by the eigenvalues of **R**, whose sum is $\mathrm{tr}(\mathbf{R}) = p$.

There is no direct relation between the results of both variants of PCA.

Correlation matrix PCA with R

With R, there are two alternative ways of performing a Correlation Matrix PCA.

PCA on standardised data

```
> prcomp(scale(lavagantes))      % or
> prcomp(lavagantes,scale=TRUE)
```

```
Standard deviations:
[1] 2.8298571 1.4518966 0.8481395 0.7315674 0.6117634 0.5371346 0.5119344 0.4730480 0.4106900
[10] 0.3761469 0.3016251 0.2178130 0.1793918

Rotation:
                    PC1          PC2          PC3         PC4         PC5         PC6         PC7
carapace_l    0.3336487 -0.051654918  0.002147496 -0.05337901  0.05903158 -0.25593010  0.13991163
tail_l        0.2328489 -0.455025510 -0.004432513  0.02919494 -0.06389168  0.06642917 -0.32471231
carapace_w    0.3399357 -0.026168964  0.042817387 -0.05649310  0.11876996 -0.18817081  0.02954496
carapace_d    0.3161771 -0.001543245  0.174339992 -0.06927295 -0.01269919  0.02103474 -0.65346959
tail_w        0.1963703 -0.522307992 -0.097172600  0.02943249  0.06817824 -0.29897195 -0.06638706
areola_l      0.2625765  0.014998718 -0.203444780 -0.78727388 -0.41920392  0.00605338  0.19498049
areola_w      0.2320279  0.063340777 -0.813027317  0.19646231  0.26234962  0.17992496 -0.10423247
rostrum_l     0.2559610 -0.260192772  0.122258123  0.50436942 -0.58565962  0.13677260  0.30765231
rostrum_w     0.3122279  0.011301755  0.084409773  0.06116672  0.43328915 -0.24980467  0.49425052
postorbital_w 0.2883485 -0.080276403  0.361940139 -0.14548391  0.36223013  0.71927271  0.11234877
propodus_l    0.2741268  0.405235606  0.006549232  0.13377738 -0.13020525 -0.02606551 -0.05259459
propodus_w    0.2474141  0.398376708  0.281998129  0.09065523  0.00717611 -0.33417966 -0.19598386
dactyl_l      0.2740158  0.339649079 -0.152524450  0.15373369 -0.22361974  0.24824297  0.03271971
[...]
```

The two variants of PCA

The results of both variants of PCA are not directly comparable.

The two variants of PCA - crayfish (lavagantes) data

```
> lav.acpR <- prcomp(lavagantes,scale=TRUE)
> summary(lav.acpR)

Importance of components:
          PC1    PC2     PC3     PC4     PC5     PC6     PC7     PC8     PC9    PC10   PC11    PC12    PC13
Std.dev 2.830 1.4519 0.84814 0.73157 0.61176 0.53713 0.51193 0.47305 0.41069 0.37615 0.3016 0.21781 0.17939
Prp.Var 0.616 0.1621 0.05533 0.04117 0.02879 0.02219 0.02016 0.01721 0.01297 0.01088 0.0070 0.00365 0.00248
Cum.Prp 0.616 0.7782 0.83350 0.87466 0.90345 0.92565 0.94581 0.96302 0.97599 0.98688 0.9939 0.99752 1.00000

> summary(lav.acp)

Importance of components:
           PC1    PC2     PC3     PC4     PC5     PC6     PC7     PC8     PC9    PC10    PC11    PC12   PC13
Std.dev 4.4171 2.1583 0.96179 0.70720 0.61636 0.49926 0.46399 0.38484 0.33629 0.25007 0.20606 0.17704 0.1406
Prp.Var 0.7265 0.1734 0.03444 0.01862 0.01415 0.00928 0.00802 0.00551 0.00421 0.00233 0.00158 0.00117 0.0007
Cum.Prp 0.7265 0.9000 0.93440 0.95302 0.96716 0.97645 0.98446 0.98998 0.99419 0.99652 0.99810 0.99926 1.0000
```

In general, a Correlation Matrix PCA needs more PCs to account for any given proportion of inertia.

The two variants of PCA (cont.)

The loadings vectors also change (eigenvectors of **S** and **R** are different), as do the vectors of scores which they produce.

Let us compute the correlations between PCs from each variant:

The two variants of PCA - `lavagantes` data (cont.)

```
> round(cor(lav.acp$x, lav.acpR$x), d=2)

       PC1   PC2   PC3   PC4   PC5   PC6   PC7   PC8   PC9  PC10  PC11  PC12  PC13
PC1   0.89  0.44  0.00  0.06 -0.07 -0.01 -0.02 -0.09 -0.04  0.04  0.01  0.04  0.03
PC2   0.44 -0.88 -0.03 -0.10  0.05 -0.05 -0.04 -0.01 -0.04  0.03 -0.05 -0.02 -0.04
PC3   0.05  0.05  0.53 -0.09  0.24 -0.42 -0.16  0.56  0.27  0.18 -0.13  0.07  0.07
PC4  -0.04 -0.10  0.19  0.79  0.09  0.11 -0.36 -0.05 -0.25  0.26  0.17  0.06  0.08
PC5   0.00  0.02 -0.03 -0.38 -0.37  0.34 -0.28  0.45 -0.42  0.31  0.21 -0.01  0.08
PC6  -0.05 -0.03 -0.21  0.02 -0.05 -0.26  0.11  0.01 -0.31 -0.05 -0.33  0.48  0.66
PC7  -0.02 -0.06 -0.26 -0.01 -0.26 -0.33 -0.12 -0.10  0.45  0.18  0.64  0.16  0.21
PC8  -0.05  0.04 -0.28  0.14 -0.27 -0.53  0.17  0.04 -0.22  0.45 -0.24 -0.15 -0.44
PC9   0.01 -0.02  0.10 -0.04  0.25  0.29  0.61 -0.08  0.10  0.64  0.08 -0.01  0.21
PC10  0.02 -0.10  0.22  0.27 -0.54  0.21  0.37  0.25  0.20 -0.17 -0.05  0.46 -0.24
PC11  0.05 -0.04 -0.04  0.27 -0.24 -0.05  0.26  0.32  0.04 -0.23  0.03 -0.68  0.40
PC12 -0.07 -0.03  0.33 -0.11 -0.47  0.07 -0.27 -0.45  0.26  0.23 -0.42 -0.18  0.21
PC13 -0.03 -0.02  0.56 -0.16 -0.13 -0.32  0.25 -0.30 -0.46 -0.15  0.38 -0.03  0.00
```

The two variants of PCA (cont.)

Correlations between the standardised data PCs and the original variables:

Correlation Matrix PCA - lavagantes data

```
> round(cor(lavagantes, lav.acpR$x), d=2)
```

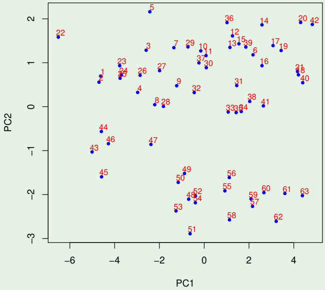
```
               PC1   PC2   PC3   PC4   PC5   PC6   PC7   PC8   PC9  PC10  PC11  PC12  PC13
carapace_l    0.94 -0.07  0.00 -0.04  0.04 -0.14  0.07 -0.11 -0.03 -0.06 -0.24  0.05 -0.04
tail_l        0.66 -0.66  0.00  0.02 -0.04  0.04 -0.17  0.05 -0.24  0.19 -0.01  0.00  0.00
carapace_w    0.96 -0.04  0.04 -0.04  0.07 -0.10  0.02 -0.10  0.08  0.07 -0.03 -0.16  0.06
carapace_d    0.89  0.00  0.15 -0.05 -0.01  0.01 -0.33  0.07 -0.02 -0.24  0.02 -0.02  0.00
tail_w        0.56 -0.76 -0.08  0.02  0.04 -0.16 -0.03 -0.18  0.17  0.00  0.12  0.07 -0.01
areola_l      0.74  0.02 -0.17 -0.58 -0.26  0.00  0.10  0.09  0.02  0.01  0.04  0.01  0.00
areola_w      0.66  0.09 -0.69  0.14  0.16  0.10 -0.05  0.14  0.08  0.01 -0.02  0.00  0.00
rostrum_l     0.72 -0.38  0.10  0.37 -0.36  0.07  0.16  0.14  0.07 -0.05 -0.01 -0.01  0.01
rostrum_w     0.88  0.02  0.07  0.04  0.27 -0.13  0.25  0.12 -0.16 -0.09  0.10  0.00 -0.01
postorbital_w 0.82 -0.12  0.31 -0.11  0.22  0.39  0.06 -0.01  0.10  0.04 -0.01  0.03  0.00
propodus_l    0.78  0.59  0.01  0.10 -0.08 -0.01 -0.03 -0.08 -0.05  0.04  0.02  0.10  0.12
propodus_w    0.70  0.58  0.24  0.07  0.00 -0.18 -0.10  0.16  0.12  0.16  0.02  0.02 -0.07
dactyl_l      0.78  0.49 -0.13  0.11 -0.14  0.13  0.02 -0.26 -0.09 -0.01  0.07 -0.03 -0.08
```

- Compared with PCA on the original data, not only do the correlations between PCs and variables change, so do possible interpretations.
- PC1 is now essentially a measure of overall size of the animal.
- PC2, is more difficult to interpret, but contrasts the size of tails and claws.

First principal plane – standardised data

Correlation matrix PCA - `lavagantes` data (cont.)

```
> plot(lav.acpR$x[,1:2],col="blue", pch=16, cex=0.8)
> text(lav.acpR$x[,1:2]+0.1, label=rownames(lavagantes), col="red")
```

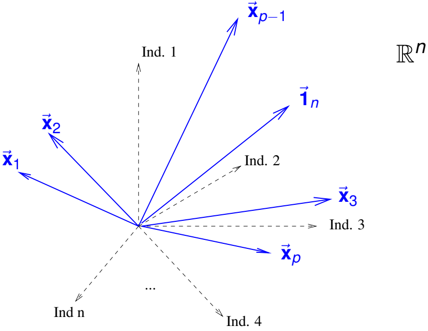


Simplifying: PC 1 orders organisms by their overall size, and PC 2 separates sex-related shape.

Representation in $\mathbb{R}^n$, the space of variables

Recall: The alternative representation of a data matrix $\mathbf{X}$, in the space of variables.

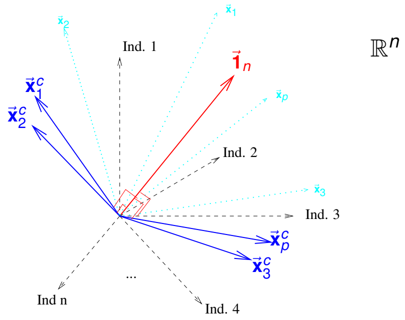
- each axis corresponds to an observed individual;
- each vector corresponds to a variable.



Centred variables in the space of variables

The most interesting representation in the space of variables is for centred variables, because geometric concepts induced by the usual inner product in $\mathbb{R}^n$ have statistical interpretations.

Centring the columns of $\mathbf{X}$ makes the vectors that represent the centred variables orthogonal to the vector $\vec{\mathbf{1}}_n$ of n ones (the sum of any column of $\mathbf{X}^c$ is zero):



Geometry and statistics in the space of variables

The generic element of the centred data matrix, $\mathbf{X}^c$, is:

$$x_{ij}^c \;=\; x_{ij} - \overline{x}_{.j} \;,$$

where

- x_{ij} denotes the observation for the i-th individual on variable j;
- $\overline{x}_{.j}$ denotes the mean of the n observations on variable j.

Thus,

- the usual norm of a column $\vec{\mathbf{x}}_j^c$ of $\mathbf{X}^c$ is proportional to that variable's standard deviation: $\|\vec{\mathbf{x}}_j^c\| = \sqrt{\sum\limits_{i=1}^{n}(x_{ij}-\overline{x}_{.j})^2} = \sqrt{n-1}\, s_j.$
- the usual inner product of two different columns of $\mathbf{X}^c$ is proportional to the covariance of those variables: $<\vec{\mathbf{x}}_j^c, \vec{\mathbf{x}}_k^c> = (\vec{\mathbf{x}}_j^c)^t\, \vec{\mathbf{x}}_k^c = \sum\limits_{i=1}^{n}(x_{ij}-\overline{x}_{.j})(x_{ik}-\overline{x}_{.k}) = (n-1)\,\mathrm{cov}_{j,k}.$
- the cosine of the angle between the vectors representing two different columns of $\mathbf{X}^c$ is the coefficient of correlation of those variables: $\cos\theta = \frac{<\vec{\mathbf{x}}_j^c, \vec{\mathbf{x}}_k^c>}{\|\vec{\mathbf{x}}_j^c\|\cdot\|\vec{\mathbf{x}}_k^c\|} = \frac{\cancel{(n-1)}\cdot\mathrm{cov}_{j,k}}{\cancel{\sqrt{n-1}}\, s_j \cdot \cancel{\sqrt{n-1}}\, s_k} = \frac{\mathrm{cov}_{j,k}}{s_j \cdot s_k} = r_{j,k}$
- Orthogonal centred vectors correspond to uncorrelated variables.

Intepretation of PCA in the space of variables

The representation in the space of variables ($\mathbb{R}^n$) associates each variable to a vector. Linear combinations of variables are linear combinations of vectors, hence new vectors. PCs are also represented by vectors in $\mathbb{R}^n$.

For centred vectors, the squared size of the vector is proportional to that variable's variance.

The PCA criterion (maximising variance) corresponds to seeking linear combinations of the vectors of maximum length (with sum of squared coefficients equal to 1).

It is geometrically intuitive that variables whose variance is much larger than others have great influence upon the first PC ("dominate the first PC").

PCA is sensitive to (different) changes of scale

Any linear (affine) transformation of a variable ($x \rightarrow a + b\,x$), as are most changes of units of measurement, re-scales the centred vector that represents it in $\mathbb{R}^n$, but preserving the direction:

- additive constants a disappear when centering, and therefore do not change the corresponding centred vector in $\mathbb{R}^n$.
- multiplicative constants b:
  - preserve the line spanned by the vector $(\vec{\mathbf{x}}_j \;\rightarrow\; b\vec{\mathbf{x}}_j)$;
  - change the direction if $b < 0$;
  - lengthen the vector if $|b| > 1$, because $\|b\vec{\mathbf{x}}_j\| = |b|\,\|\vec{\mathbf{x}}_j\|$;
  - shorten the vector if $|b| < 1$.

Thus, the PCA criterion is sensitive to different changes of scale in the p variables.

Interpretation of PCA in $\mathbb{R}^n$ (cont.)

What is the effect of standardising the variables on the representation in $\mathbb{R}^n$?

Standardising the data (as in a correlation matrix PCA) makes all vectors representing the centred variables equal in size.

Thus,

- there will not be vectors that are larger than others, unduly influencing the first PCs;
- what will essentially determine the direction of greatest length is the pattern of correlations among the variables, i.e., their relative angular position;
- groups of strongly correlated variables tend to “attract” the first PC of the standardised data.

More on Correlation Matrix PCA

In geometric terms, standardising the variables:

- In $\mathbb{R}^n$, re-sizes each of the p vectors, to a common size (norm).
- In $\mathbb{R}^p$, stretches or compresses each axis, with re-scaling factors that are different for each axis. It changes the shape of the scatterplot.

Observations:

- The total variability is $\mathrm{tr}(\mathbf{R}) = p$ (the number of variables).
- The correlation between variable $\vec{\mathbf{x}}_i$ and the j-th PC is now $\sqrt{\lambda_j^R}\, v_{ij}^R$.
- Sometimes, the loadings in a correlation matrix PCA are rescaled so that $\vec{\mathbf{v}}_j^t \vec{\mathbf{v}}_j = \lambda_j$. In that case, the new loadings of the linear combination are the correlations between the variable and the PC.

Warnings about PCA (in general)

- Reducing dimensionality with PCA does not mean reducing the number of original variables: each PC is a linear combination of all the observed variables.
- Each PC is often interpreted ignoring the variables whose loadings in the linear combination defining the PC are “close to zero”. This may mislead, and additional information should be used to validate loadings-based interpretations.
- Another frequent, but debatable, practice in PCA is the rotation of PCs: loadings are changed to make them closer to zero or one, with a view to “simplifying the interpretation”. But this goal may be illusory (as we saw) and sacrifices the optimality of the solutions.
- Some authors also call the eigenvectors of **S** or **R** (loadings vectors) principal components, sowing confusion.
- It does not make sense to use factors (qualitative or categorical variables) in the data.

An alternative approach to PCA

Principal Component Analysis can also be introduced with the fundamental result of Matrix Theory: the Singular Value Decomposition (SVD).

As with the Spectral Decomposition, the SVD involves the factorisation of a matrix into the product of 3 matrices, with the central matrix being diagonal and the two others having orthonormal columns. But:

- While the Spectral Decomposition is only valid for symmetric matrices, the SVD is valid for any matrix, including rectangular matrices.
- The three matrices of an SVD are different and, in general, are of different sizes.
- The SVD and the Spectral Decomposition coincide in the case of symmetric matrices with non-negative eigenvalues.

Singular Value Decomposition

Singular Value Decomposition (SVD)

Let $\mathbf{Y}_{n\times p}$ be a generic matrix. It is always possible to factorise $\mathbf{Y}$ as follows:

$$\mathbf{Y} = \mathbf{W\Delta V}^t \quad \Longleftrightarrow \quad \mathbf{Y} = \sum_{i=1}^{p} \delta_i \vec{\mathbf{w}}_i \vec{\mathbf{v}}_i^t \ ,$$

where

$\mathbf{\Delta}_{p\text{x}p}$	diagonal matrix
$\mathbf{V}_{p\text{x}p}$	matrix with orthonormal columns ($\mathbf{V}^t\mathbf{V} = \mathbf{I}_p$)
$\mathbf{W}_{n\text{x}p}$	matrix with orthonormal columns ($\mathbf{W}^t\mathbf{W} = \mathbf{I}_p$)
δ_i	diagonal elements of $\mathbf{\Delta}$ (singular values of $\mathbf{Y}$)
$\vec{\mathbf{w}}_i$	columns of $\mathbf{W}$ (left singular vectors of $\mathbf{Y}$)
$\vec{\mathbf{v}}_i$	columns of $\mathbf{V}$ (right singular vectors of $\mathbf{Y}$)

We assume that the singular values δ_i are in decreasing order.

Observations on the SVD: $\mathbf{Y} = \mathbf{W}\boldsymbol{\Delta}\mathbf{V}^t$

- The transpose $\mathbf{Y}^t$ has Singular Value Decomposition $\mathbf{Y}^t = \mathbf{V}\boldsymbol{\Delta}\mathbf{W}^t$.
- $\mathbf{Y}^t\mathbf{Y} = \mathbf{V}\boldsymbol{\Delta}^2\mathbf{V}^t$ is a Spectral Decomposition of $\mathbf{Y}^t\mathbf{Y}$. Hence, $\mathbf{V}$ is a matrix whose columns are an orthonormal set of eigenvectors of $\mathbf{Y}^t\mathbf{Y}$.
- $\mathbf{W}$ is an analogous matrix, of eigenvectors of $\mathbf{Y}\mathbf{Y}^t = \mathbf{W}\boldsymbol{\Delta}^2\mathbf{W}^t$.
- $\boldsymbol{\Delta}$ is the diagonal matrix of square roots of the eigenvalues of $\mathbf{Y}^t\mathbf{Y}$ (which, if non-zero, are also eigenvalues of $\mathbf{Y}\mathbf{Y}^t$).
- The SVD of a matrix is always possible, though not unique (ate least due to sign-switching in pairs of vectors).
- If $\mathbf{Y}$ has rank (maximum number of linearly independent columns) $r < p$, then $\delta_i = 0$ for $i > r$.
- If $\mathbf{Y}$ has rank $r < p$, the $p - r$ final terms in the sum $\mathbf{Y} = \sum_{i=1}^{p} \delta_i \vec{\mathbf{w}}_i \vec{\mathbf{v}}_i^t$ are matrices of zeros. This means that the $p - r$ final columns of $\mathbf{V}$ and $\mathbf{W}$, and the $p - r$ final rows/columns of $\boldsymbol{\Delta}$ can be dropped. The resulting SVD is called the Thin SVD.

SVD and PCA

PCA corresponds to a Singular Value Decomposition of a centred data matrix $\mathbf{X}^{\mathrm{c}}$, divided by $\sqrt{n-1}$, (or $\sqrt{n}$, depending on the convention used to define covariances):

$$\frac{1}{\sqrt{n-1}}\mathbf{X}^{\mathrm{c}} = \mathbf{U}\boldsymbol{\Delta}\mathbf{V}^{t} \,,$$

with:

$\mathbf{V}$ - matrix whose columns are eigenvectors of $\mathbf{S} = \frac{1}{n-1}\mathbf{X}^{\mathrm{c}\,t}\mathbf{X}^{\mathrm{c}}$, that is, with PC loadings.

$\boldsymbol{\Delta}$ - matrix whose diagonal elements are square roots of eigenvalues of $\mathbf{S}$, i.e., standard deviations of the PCs;

$\mathbf{X}^{\mathrm{c}}\mathbf{V} = \sqrt{n-1}\,\mathbf{U}\boldsymbol{\Delta}$ - matrix whose columns are centred scores for the individuals on each PC.

$\mathbf{U} = \frac{1}{\sqrt{n-1}}\mathbf{X}^{\mathrm{c}}\mathbf{V}\Delta^{-1}$ - matrix of left singular vectors, which are vectors of normalised scores.

SVD and PCA (cont.)

We illustrate, carrying out the SVD of a matrix $\frac{1}{\sqrt{n-1}}\mathbf{X}^c$ with R, for the crayfish (lavagantes) dataset.

Centring a data matrix can be done as follows:

```
> lav.centrado <- scale(lavagantes, scale=FALSE)
```

The command scale can both centre (subtract the means) and divide by standard deviations of the matrix columns.

Each of these operations is controlled by an argument, respectively center and scale.

By default, these arguments are TRUE. Any of these operation may be omitted setting the corresponding argument to the logical value FALSE.

In R, a Singular Value Decomposition is done with the command svd.

PCA and SVD (cont.)

SVD with crayfish data

```
> svd(lav.centrado/sqrt(62))

$d
[1] 4.4171243 2.1583124 0.9617894 0.7071970 0.6163559 0.4992560 0.4639879
[8] 0.3848417 0.3362918 0.2500701 0.2060563 0.1770375 0.1405790

$u
             [,1]         [,2]         [,3]          [,4]         [,5]
[1,] -0.144379990 -0.182396510 -0.123645871  0.1059842750  0.070557452
[2,] -0.144331125 -0.157779185  0.254967864  0.1172558607  0.146926297
[3,] -0.059725146 -0.177923620 -0.059333869  0.1101757182  0.113206688
[4,] -0.093246935 -0.113657051  0.014976742  0.0804924915 -0.069971697
[5,] -0.035380664 -0.210254166 -0.097758921 -0.1206499751 -0.146049537
[...]

$v
           [,1]        [,2]        [,3]        [,4]         [,5]         [,6]
[1,] 0.28762060  0.36935786  0.08475822 -0.31404094 -0.454639049  0.272071976
[2,] 0.10615292  0.61487598 -0.01728674  0.46421995  0.550775374  0.088028646
[3,] 0.19089393  0.22112280  0.09978650 -0.10987953 -0.186701149 -0.178125878
[4,] 0.13951311  0.14784642  0.13138041  0.01598041  0.105009202 -0.171612241
[5,] 0.04682070  0.49290700 -0.05172379  0.06592005 -0.405755003 -0.046182873
[...]
```

Warning: Output components `$u` and `$v` are, respectively, the matrices $\mathbf{U}$ and $\mathbf{V}$.

Component `$d` is a vector, with the diagonal elements of matrix $\boldsymbol{\Delta}$.

PCA and SVD (cont.)

SVD for the crayfish (cont.)

```
> DVS <- svd(lav.centrado/sqrt(62))
> U <- DVS$u
> D <- diag(DVS$d)    <- creates a diagonal matrix from vector DVS$d
> U %*% D * sqrt(62)
```

```
            [,1]        [,2]        [,3]         [,4]        [,5]         [,6]
[1,]  -5.0216041 -3.09975004 -0.93638716  0.590170762  0.34242883 -0.311295721
[2,]  -5.0199046 -2.68138921  1.93090666  0.652936303  0.71306147  2.411219117
[3,]  -2.0772687 -3.02373521 -0.44934354  0.613510708  0.54941375 -0.365822245
[...]
[62,]  1.5767872  4.68339718 -0.49231884  0.246787192 -0.11313707  0.138658304
[63,]  3.2782407  4.30830749  0.15373020 -0.562657698 -0.73379507  0.200035217
[...]
```

The command `prcomp` uses the SVD:

```
> prcomp(lavagantes)$x
```

```
          PC1         PC2         PC3          PC4         PC5          PC6
1  -5.0216041 -3.09975004 -0.93638716  0.590170762  0.34242883 -0.311295721
2  -5.0199046 -2.68138921  1.93090666  0.652936303  0.71306147  2.411219117
3  -2.0772687 -3.02373521 -0.44934354  0.613510708  0.54941375 -0.365822245
[...]
62  1.5767872  4.68339718 -0.49231884  0.246787192 -0.11313707  0.138658304
63  3.2782407  4.30830749  0.15373020 -0.562657698 -0.73379507  0.200035217
[...]
```

(No) A geometric problem

A data matrix $\mathbf{X}_{n\times p}$ is represented by an n-point scatterplot in $\mathbb{R}^p$ or, alternatively, a bundle of p vectors in $\mathbb{R}^n$.

If $\mathbf{Y}_{n\times p}$ is a matrix of equal size, but rank $r < p$, the corresponding n-point scatterplot is on a subspace of dimension r of $\mathbb{R}^p$. Likewise, its bundle of p vectors spans a subspace of dimension r in $\mathbb{R}^n$.

Geometric problem

To identify the matrix $\mathbf{Y}_{n\times p}$, of rank r, whose n points in $\mathbb{R}^p$ minimise the sum of squares of the distances to the n points associated with the original data matrix $\mathbf{X}_{n\times p}$:

$$\sum_{i=1}^{n}\sum_{j=1}^{p}\left(x_{ij}-y_{ij}\right)^2 \ .$$

This criterion also minimises the sum of squared distances between the p columns of **X** and **Y**, so that the p-vector bundle defined by **Y** is “the closest, overall” to the p vectors defined by **X**.

(No) The solution

Eckart-Young Theorem

Let $\mathbf{X}_{n\text{x}p}$ be a matrix of rank p. The matrix $\mathbf{Y}_{n\text{x}p}$ of rank $r < p$ that minimises the usual matrix distance $\|\mathbf{X}-\mathbf{Y}\| = \sqrt{\sum_i \sum_j (x_{ij}-y_{ij})^2}$, is obtained as follows:

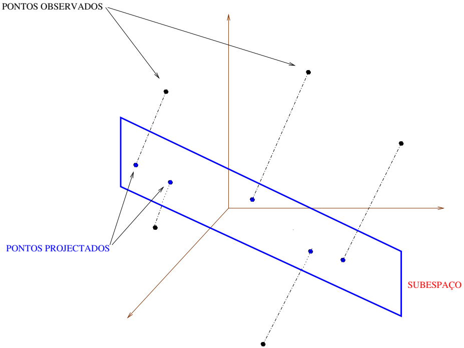
- Let $\mathbf{X} = \mathbf{W\Delta V}^t$ be the singular value decomposition of $\mathbf{X}$.
- Let $\mathbf{W}_r$, $\mathbf{V}_r$, be the matrices of r columns of $\mathbf{W}$ and $\mathbf{V}$, respectively, associated with the r largest singular values.
- Let $\mathbf{\Delta}_r$ be the diagonal matrix of size $r \times r$ resulting from retaining only the r largest singular values of $\mathbf{\Delta}$.
- Then $\mathbf{Y} = \mathbf{W}_r\mathbf{\Delta}_r\mathbf{V}_r^t$ (and this is an SVD of $\mathbf{Y}$).

Note 1: If $\mathbf{X} = \sum_{i=1}^{p} \delta_i \vec{\mathbf{w}}_i \vec{\mathbf{v}}_i^t$ is the SVD of $\mathbf{X}$, $\mathbf{Y}$ is the matrix that results from retaining only the first r terms of that sum: $\mathbf{Y} = \sum_{i=1}^{r} \delta_i \vec{\mathbf{w}}_i \vec{\mathbf{v}}_i^t$.

Note 2: Thus, PCA (the SVD of $\mathbf{X}^c$) identifies, both in $\mathbb{R}^p$ and in $\mathbb{R}^n$, the subspaces of dimension r where the representation of the data is as faithful as possible, in the sense of being the closest to the original values.

(No) Orthogonal projections in $\mathbb{R}^p$ and $\mathbb{R}^n$

For both representations of the data from $\mathbf{X}^c$, PCA solves the problem of identifying the subspace of dimension r where the orthogonal projection of the data onto that subspace minimises the sum of squared (perpendicular) distances between original and projected points.



Biplots

- Intimately connected with the Singular Value Decomposition of a centred data matrix (therefore, with a PCA).
- Fundamental ideia in a *biplot*: obtain a good low-dimensional (approximate) representation of both the individuals and the variables (hence the prefix *bi*-).
- geometrically preserving the main statistical characteristics of the data.

Biplots (cont.)

- Let $\mathbf{X}^c$ be a centred data matrix, with SVD: $\frac{1}{\sqrt{n-1}}\mathbf{X}^c = \mathbf{U\Delta V}^t$.
- Defining:

$$\begin{aligned} \mathbf{G} &= \mathbf{U} \\ \mathbf{H} &= \mathbf{V\Delta} \end{aligned}$$

  we have: $\frac{1}{\sqrt{n-1}}\mathbf{X}^c = \mathbf{GH}^t$.

- If $\mathbf{X}^c$ is of rank p,
  - $\mathbf{G}$ is $n \times p$ and the rows of $\mathbf{G}$ correspond to individuals.
  - $\mathbf{H}$ is $p \times p$ and the rows of $\mathbf{H}$ correspond to variables.
- The rows of $\mathbf{G}$ ($g^t_{[i]}$) and of $\mathbf{H}$ ($h^t_{[j]}$) are markers for, respectively, individuals and variables, which belong to the same space ($\mathbb{R}^p$) and can be represented together.
- The inner product of the markers for individual i and for variable j is the value for that individual on that variable (centred and divided by $\sqrt{n-1}$):

$$g^t_{[i]}h_{[j]} \; = \; \frac{1}{\sqrt{n-1}}x^c_{ij} \; .$$

Variable markers

Consider the properties of variable markers, which are vectors in $\mathbb{R}^p$. The inner products of variable markers are:

$$\mathbf{H}\mathbf{H}^t = (\mathbf{V}\boldsymbol{\Delta})(\mathbf{V}\boldsymbol{\Delta})^t = \mathbf{V}\boldsymbol{\Delta}^2\mathbf{V}^t = \mathbf{S} \, ,$$

since $\mathbf{S} = \frac{1}{n-1}\mathbf{X}^{c\,t}\mathbf{X}^c = (\mathbf{U}\boldsymbol{\Delta}\mathbf{V}^t)^t(\mathbf{U}\boldsymbol{\Delta}\mathbf{V}^t) = \mathbf{V}\boldsymbol{\Delta}\mathbf{U}^t\mathbf{U}\boldsymbol{\Delta}\mathbf{V}^t = \mathbf{V}\boldsymbol{\Delta}^2\mathbf{V}^t$.

- The inner product of markers for pairs of variables give the covariance between those variables.
- The norm (size) of each variable marker is the standard deviation of that variable.
- The cosine of the angle between each pair of variable markers is the coefficient of linear correlation between the variables.

Mahalanobis distances

To understand the properties of markers for individuals, we must introduce (squared) Mahalanobis distances.

Mahalanobis distances

Let $\mathbf{X}_{n\times p}$ be a data matrix, with generic row $\vec{\mathbf{x}}_{[i]}$, covariance matrix $\mathbf{S}$ and centre of gravity $\vec{\mathbf{m}}$. Define:

- the (squared) Mahalanobis distance of individual i to the centre:

$$\|\vec{\mathbf{x}}_{[i]} - \vec{\mathbf{m}}\|^2_{\mathbf{S}^{-1}} = (\vec{\mathbf{x}}_{[i]} - \vec{\mathbf{m}})^t \mathbf{S}^{-1} (\vec{\mathbf{x}}_{[i]} - \vec{\mathbf{m}}) \ .$$

- the (squared) Mahalanobis distance between individuals i and j:

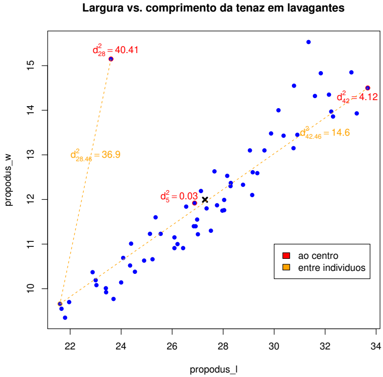
$$\|\vec{\mathbf{x}}_{[i]} - \vec{\mathbf{x}}_{[j]}\|^2_{\mathbf{S}^{-1}} = (\vec{\mathbf{x}}_{[i]} - \vec{\mathbf{x}}_{[j]})^t \mathbf{S}^{-1} (\vec{\mathbf{x}}_{[i]} - \vec{\mathbf{x}}_{[j]}) \ .$$

The usual Euclidean distances are given by similar expressions, but with the identity matrix $\mathbf{I}$ in place of the matrix $\mathbf{S}^{-1}$.

Mahalanobis distances (cont.)

Mahalanobis distances take into account the shape of the scatterplot in $\mathbb{R}^p$ (pattern of covariances between variables). They can be useful in identifying multivariate outliers.

This is the scatterplot in $\mathbb{R}^2$ for the crayfish variables `propodus_l` and `propodus_w`:



Largura vs. comprimento da tenaz em lavagantes

The centre of gravity is marked by a black cross.

The numerical values are Mahalanobis distances.

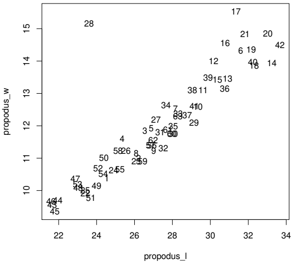
In a biplot, the Euclidean distance between markers for individuals is equal to the Mahalanobis distance between the individuals.

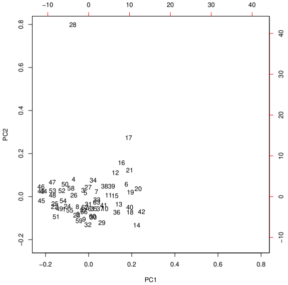
Markers for individuals

- The Euclidean distance between each pair of rows of **G** is proportional to the Mahalanobis distance between the corresponding individuals:

$$\|\vec{\mathbf{g}}_{[i]}-\vec{\mathbf{g}}_{[j]}\|^2 \;=\; (\vec{\mathbf{x}}_{[i]}-\vec{\mathbf{x}}_{[j]})^t\mathbf{S}^{-1}(\vec{\mathbf{x}}_{[i]}-\vec{\mathbf{x}}_{[j]}) \;=\; \|\vec{\mathbf{x}}_{[i]}-\vec{\mathbf{x}}_{[j]}\|^2_{\mathbf{S}^{-1}} \;.$$

Here is the scatterplot of the variables `propodus_l` and `propodus_w` and their biplot markers for individuals:





In the biplot, the Euclidean distance between points is the Mahalanobis distance between individuals.

Biplots (cont.)

The visualization of a biplot requires reducing the representation to a $k = 2$ or $k = 3$ dimensional space.

This is done by retaining only the marker coordinates for the first two (or three) dimensions:

- $\mathbf{G}^{(k)}$ $n \times k$ submatrix with the first k columns of $\mathbf{G}$.
- $\mathbf{H}^{(k)}$ $p \times k$ submatrix with the first k columns of $\mathbf{H}$.

The rows of $\mathbf{G}^{(k)}$ and $\mathbf{H}^{(k)}$ are markers for individuals and variables and:

$$\frac{1}{\sqrt{n-1}}\tilde{\mathbf{X}}^{\mathrm{c}} = \mathbf{G}^{(k)}{\mathbf{H}^{(k)}}^{t}$$

is the best rank k approximation of $\frac{1}{\sqrt{n-1}}\mathbf{X}^{\mathrm{c}}$ (Eckart-Young Theorem).

Biplots (cont.)

By taking $k = 2$, we get a 2-dimensional scatterplot, with

- markers for individuals represented by points; and
- markers for variables represented by vectors.

We have, approximately:

- the cosine of the angle between variable markers is the coefficient of correlation between variables;
- the length of each variable marker is proportional to its standard deviation;
- the Euclidean distance between individual markers is the Mahalanobis distance between those individuals:

$$M_{ij} \ = \ (\vec{\mathbf{x}}_{[i]} - \vec{\mathbf{x}}_{[j]})^t \mathbf{S}^{-1} (\vec{\mathbf{x}}_{[i]} - \vec{\mathbf{x}}_{[j]}) \ ,$$

The quality of this approximation can be measured as in PCA.

Biplots (cont.)

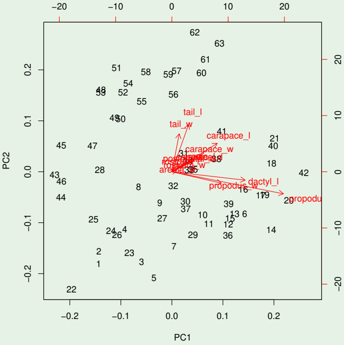
We also have the following approximate (only $k = 2$ dimensions) properties:

- the cosine of the angle between each vector and the horizontal axis is approximately the coefficient of linear correlation between that variable and PC 1;
- the cosine of the angle between each vector and the vertical axis is approximately the coefficient of linear correlation between each variable and PC 2;
- The orthogonal projection of each point on the line defined by each vector is approximately the value of each individual on the corresponding variable.

Biplots with R

The `biplot` command for the crayfish data

```
> biplot(lav.acp)
```



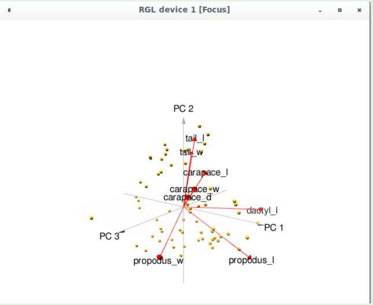
The individuals in group 43-63 (females) tend to have smaller claws and larger tails than the males.

3*D* biplots with pca3d

Three-dimensional biplots with package pca3d

Add to the command pca3d the argument biplot=TRUE:

```
> library(pca3d)
> pca3d(lav.acp, biplot=TRUE)
```

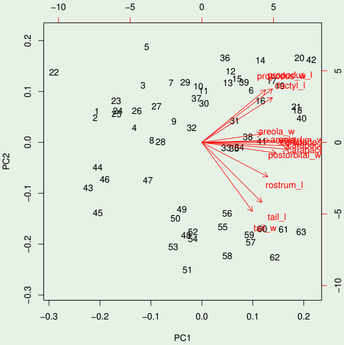


The picture has frozen one moment in the rotation. By default, not all variable markers are shown.

Biplots with R (cont.)

Function `biplot` for the standardised crayfish data

```
> biplot(lav.acpR)
```

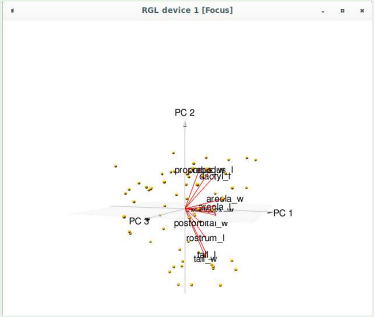


The male/female separation is still visible. The first PC now points in the direction of a highly correlated group of variables (size).

A 3*D* biplot for the standardised crayfish data

The 3-D biplot using package pca3d

```
> pca3d(lav.acpR, biplot=TRUE, biplot.vars=13)
```



The argument biplot.vars provides control over the variable markers that are shown.

Linear Discriminant Analysis

Discriminant Analyses

Multivariate methods that:

- Assume that the n observed individuals belong to k (known) subgroups or classes.
- Seek functions of the p observed variables that best distinguish or discriminate those subgroups.

Linear (or Fisher's) Discriminant Analysis:

Seeks linear combinations of the p observed variables which best discriminate the subgroups.

NOTE: We assume a descriptive context, although often Discriminant Analyses are introduced with probabilistic concepts.

Linear Discriminant Analysis (cont.)

Starting point: a data matrix $\mathbf{X}_{n \times p}$.

The n individuals (rows of $\mathbf{X}$) define a partition into k subgroups, that is known. They can be seen as k factor levels.

Informal goal: determine the best linear combination $\mathbf{X}\vec{\mathbf{a}}$ of observed variables that can ensure that:

- individuals of a common class have similar values, and
- individuals in different classes have values far apart.

Solutions: linear combinations $\mathbf{X}\vec{\mathbf{a}}$, called discriminant axes (or sometimes canonical variables).

The solution involves orthogonal projections on the subspace of $\mathbb{R}^n$ spanned by the indicator variables for each subgroup (the same as in a one-way ANOVA).

The classification matrix

The classification matrix **G**, whose *i*-th column is the indicator variable for subgroup *i*:

$$\mathbf{G}_{n\times k} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \cdots & 0 \\ -- & -- & -- & -- & -- \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & 0 & \cdots & 0 \\ -- & -- & -- & -- & -- \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -- & -- & -- & -- & -- \\ 0 & 0 & 0 & \cdots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

It is similar to the one-way ANOVA model matrix, but the first column is the indicator variable for factor level 1. The subspace of $\mathbb{R}^n$ spanned by two matrices is the same.

There is a close connection between LDA and a one-way ANOVA, although we use only descriptive concepts when defining LDA.

The classification matrix (cont.)

The vectors of the column-space of matrix $\mathbf{G}$ have the same value for all elements in each subgroup, that is, $\vec{\mathbf{z}} \in \mathscr{C}(\mathbf{G})$ are of the form:

$$\vec{\mathbf{z}}^t \quad = \quad [\, \underbrace{z_1 \;\; z_1 \;\; \ldots \;\; z_1}_{n_1 \text{ vezes}} \,|\, \underbrace{z_2 \;\; z_2 \;\; \ldots \;\; z_2}_{n_2 \text{ vezes}} \,|\, \cdots \,|\, \underbrace{z_k \;\; z_k \;\; \ldots \;\; z_k}_{n_k \text{ vezes}} \,]$$

Hence, vectors in $\mathscr{C}(\mathbf{G})$ are homogeneous within classes.

But not necessarily heterogeneous between classes: $\mathscr{C}(\mathbf{G})$ also includes the vector $\vec{\mathbf{1}}_n$, which does not discriminate subgroups.

Maximising heterogeneity between classes means maximising the variability of the k values $\{z_j\}_{j=1}^k$.

We would like the linear combination to be as far away as possible from $\mathscr{C}(\vec{\mathbf{1}}_n) \subset \mathscr{C}(\mathbf{G})$, say orthogonal to vector $\vec{\mathbf{1}}_n$.

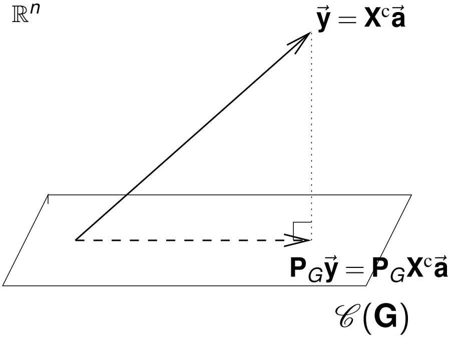
Formulation of the problem

Vectors orthogonal to vector $\vec{\mathbf{1}}_n$ are centred vectors.

Consider only the centred linear combinations: $\mathbf{X}^c\vec{\mathbf{a}}$

The orthogonal projection of any centred linear combination on the column-space of the classification matrix $\mathbf{G}$ is $\mathbf{P}_G\mathbf{X}^c\vec{\mathbf{a}}$, where $\mathbf{P}_G = \mathbf{G}(\mathbf{G}^t\mathbf{G})^{-1}\mathbf{G}^t$.

The orthogonal projection creates a right triangle:



$\mathbb{R}^n$
$\vec{\mathbf{y}} = \mathbf{X}^c\vec{\mathbf{a}}$
$\mathbf{P}_G\vec{\mathbf{y}} = \mathbf{P}_G\mathbf{X}^c\vec{\mathbf{a}}$
$\mathscr{C}(\mathbf{G})$

Revisiting Pythagoras

By the Pythagorean Theorem, and since $\mathbf{P}_G$ are $\mathbf{I}_n$ are symmetric and idempotent, we have:

$$\begin{array}{rrcl} & \|\mathbf{X}^{\rm c}\vec{\mathbf{a}}\|^2 & = & \|\mathbf{P}_G\mathbf{X}^{\rm c}\vec{\mathbf{a}}\|^2 + \|(\mathbf{I}_n - \mathbf{P}_G)\mathbf{X}^{\rm c}\vec{\mathbf{a}}\|^2 \\ \Leftrightarrow & \vec{\mathbf{a}}^t\mathbf{X}^{{\rm c}\,t}\mathbf{X}^{\rm c}\vec{\mathbf{a}} & = & \vec{\mathbf{a}}^t\mathbf{X}^{{\rm c}\,t}\mathbf{P}_G\mathbf{X}^{\rm c}\vec{\mathbf{a}} + \vec{\mathbf{a}}^t\mathbf{X}^{{\rm c}\,t}(\mathbf{I}_n - \mathbf{P}_G)\mathbf{X}^{\rm c}\vec{\mathbf{a}} \end{array}$$

The left-hand side of the equation is $\vec{\mathbf{a}}^t\mathbf{X}^{{\rm c}\,t}\mathbf{X}^{\rm c}\vec{\mathbf{a}} = (n-1)\cdot\vec{\mathbf{a}}^t\mathbf{S}\vec{\mathbf{a}}$, i.e., $n-1$ times the variance of the linear combination $\mathbf{X}^{\rm c}\vec{\mathbf{a}}$.

The desirable linear combination $\mathbf{X}^{\rm c}\vec{\mathbf{a}}$ will, in this decomposition, maximise (in relative terms) the first term on the right-hand side: this maximises the variability of the class values z_j.

The matrix of orthogonal projections $\mathbf{P}_\mathbf{G}$ (cont.)

Matrix $\mathbf{P}_G = \mathbf{G}(\mathbf{G}^t\mathbf{G})^{-1}\mathbf{G}^t$ is a block-diagonal matrix with a single value in each diagonal block: $\frac{1}{n_i}$.

$$\mathbf{P}_G = \left[\begin{array}{c|c|c|c}
\begin{array}{cccc} \frac{1}{n_1} & \frac{1}{n_1} & \cdots & \frac{1}{n_1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{n_1} & \frac{1}{n_1} & \cdots & \frac{1}{n_1} \end{array} & \mathbf{0}_{n_1 \mathrm{x} n_2} & \cdots & \mathbf{0}_{n_1 \mathrm{x} n_k} \\ \hline
\mathbf{0}_{n_2 \mathrm{x} n_1} & \begin{array}{cccc} \frac{1}{n_2} & \frac{1}{n_2} & \cdots & \frac{1}{n_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{n_2} & \frac{1}{n_2} & \cdots & \frac{1}{n_2} \end{array} & \cdots & \mathbf{0}_{n_2 \mathrm{x} n_k} \\ \hline
\vdots & \vdots & \ddots & \vdots \\ \hline
\mathbf{0}_{n_k \mathrm{x} n_1} & \mathbf{0}_{n_k \mathrm{x} n_2} & \cdots & \begin{array}{cccc} \frac{1}{n_k} & \frac{1}{n_k} & \cdots & \frac{1}{n_k} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{n_k} & \frac{1}{n_k} & \cdots & \frac{1}{n_k} \end{array}
\end{array}\right]$$

Pre-multiplying any vector by $\mathbf{P}_G$, replaces each element of the vector by its group (factor level) mean.

The projected vectors $\mathbf{P}_G\vec{\mathbf{y}}$

Consider any vector $\vec{\mathbf{y}} \in \mathbb{R}^n$, with doubly-indexed elements (i,j), where i denotes subroup and j repetition. Consider also its orthogonal projection onto $\mathscr{C}(\mathbf{G})$:

$$\vec{\mathbf{y}} = \left[\begin{array}{c} y_{11} \\ \vdots \\ y_{1\,n_1} \\ \hline y_{21} \\ \vdots \\ y_{2\,n_2} \\ \hline \vdots \\ \hline y_{k1} \\ \vdots \\ y_{k\,n_k} \end{array}\right] \qquad \mathbf{P}_G\vec{\mathbf{y}} = \left[\begin{array}{c} \overline{y}_{1.} \\ \vdots \\ \overline{y}_{1.} \\ \hline \overline{y}_{2.} \\ \vdots \\ \overline{y}_{2.} \\ \hline \vdots \\ \hline \overline{y}_{k.} \\ \vdots \\ \overline{y}_{k.} \end{array}\right]$$

Projected centred vectors $\mathbf{P}_G\vec{\mathbf{y}}^c$

Now consider a centred vector:

$$\vec{\mathbf{y}}^c = \left[\begin{array}{c} y_{11} - \overline{y}_{..} \\ \vdots \\ y_{1\,n_1} - \overline{y}_{..} \\ \hline y_{21} - \overline{y}_{..} \\ \vdots \\ y_{2\,n_2} - \overline{y}_{..} \\ \hline \vdots \\ \hline y_{k1} - \overline{y}_{..} \\ \vdots \\ y_{k\,n_k} - \overline{y}_{..} \end{array}\right] \qquad \mathbf{P}_G\vec{\mathbf{y}}^c = \left[\begin{array}{c} \overline{y}_{1.} - \overline{y}_{..} \\ \vdots \\ \overline{y}_{1.} - \overline{y}_{..} \\ \hline \overline{y}_{2.} - \overline{y}_{..} \\ \vdots \\ \overline{y}_{2.} - \overline{y}_{..} \\ \hline \vdots \\ \hline \overline{y}_{k.} - \overline{y}_{..} \\ \vdots \\ \overline{y}_{k.} - \overline{y}_{..} \end{array}\right]$$

$\|\mathbf{P}_G\vec{\mathbf{y}}^c\|^2 = \sum_{i=1}^{k}\sum_{j=1}^{n_i}(\overline{y}_{i.} - \overline{y}_{..})^2$ measures the dispersion of class means around the overall mean $\overline{y}_{..}$. It is the Factor Sum of Squares *SQF* in the one-way ANOVA of $\vec{\mathbf{y}}$ on the factor defining the classes. It is between-class variability, and should be large, as it reflects heterogeneity between classes.

The vectors $(\mathbf{I}_n - \mathbf{P}_G)\vec{\mathbf{y}}$

For any vector $\vec{\mathbf{y}} \in \mathbb{R}^n$, including centred vectors $\vec{\mathbf{y}}^c$:

$$\vec{\mathbf{y}} - \mathbf{P}_G\vec{\mathbf{y}} = (\mathbf{I}_n - \mathbf{P}_G)\vec{\mathbf{y}} = \left[\begin{array}{c} y_{11} - \overline{y}_{1.} \\ \vdots \\ y_{1n_1} - \overline{y}_{1.} \\ \hline y_{21} - \overline{y}_{2.} \\ \vdots \\ y_{2n_2} - \overline{y}_{2.} \\ \hline \vdots \\ \hline y_{k1} - \overline{y}_{k.} \\ \vdots \\ y_{kn_k} - \overline{y}_{k.} \end{array}\right] \qquad (\mathbf{I}_n - \mathbf{P}_G)\vec{\mathbf{y}}^c = \left[\begin{array}{c} y_{11} - \overline{y}_{1.} \\ \vdots \\ y_{1n_1} - \overline{y}_{1.} \\ \hline y_{21} - \overline{y}_{2.} \\ \vdots \\ y_{2n_2} - \overline{y}_{2.} \\ \hline \vdots \\ \hline y_{k1} - \overline{y}_{k.} \\ \vdots \\ y_{kn_k} - \overline{y}_{k.} \end{array}\right]$$

$\|(\mathbf{I}_n - \mathbf{P}_G)\vec{\mathbf{y}}^c\|^2 = \sum_{i=1}^{k} \sum_{j=1}^{n_i} (y_{ij} - \overline{y}_{i.})^2$ measures the dispersion of individual observations around their class (level) mean. It is the Residual Sum of Squares, *SQRE*, in the one-way ANOVA of $\vec{\mathbf{y}}$ on the classification factor. It is within-class variability and should be small: it reflects class homogeneity.

Again the equation from Pythagoras

The equation on slide 88 simplifies if we define the matrices:

$$\begin{array}{rl} \mathbf{S} = \frac{1}{n-1}\mathbf{X}^{c\,t}\mathbf{X}^{c} & \text{(Co)variance matrix for } \mathbf{X} \\ \mathbf{B} = \frac{1}{n-1}\mathbf{X}^{c\,t}\mathbf{P}_G\mathbf{X}^{c} & \text{Matrix of between-class variability} \\ \mathbf{W} = \frac{1}{n-1}\mathbf{X}^{c\,t}(\mathbf{I}_n - \mathbf{P}_G)\mathbf{X}^{c} & \text{Matrix of within-class variability} \end{array}$$

We have:

$$\vec{\mathbf{a}}^t \underbrace{\mathbf{X}^{c\,t}\mathbf{X}^{c}}_{=(n-1)\cdot\mathbf{S}} \vec{\mathbf{a}} = \underbrace{\vec{\mathbf{a}}^t\mathbf{X}^{c\,t}\mathbf{P}_G\mathbf{X}^{c}\vec{\mathbf{a}}}_{=(n-1)\cdot\mathbf{B}} + \underbrace{\vec{\mathbf{a}}^t\mathbf{X}^{c\,t}(\mathbf{I}_n - \mathbf{P}_G)\mathbf{X}^{c}\vec{\mathbf{a}}}_{=(n-1)\cdot\mathbf{W}}$$

$$\Longleftrightarrow \qquad \vec{\mathbf{a}}^t\mathbf{S}\vec{\mathbf{a}} = \vec{\mathbf{a}}^t\mathbf{B}\vec{\mathbf{a}} + \vec{\mathbf{a}}^t\mathbf{W}\vec{\mathbf{a}}$$

Fisher's formulation of the problem: among all possible linear combinations $\mathbf{X}^{c}\vec{\mathbf{a}}$, choose that which maximises:

$$\frac{\vec{\mathbf{a}}^t\mathbf{B}\vec{\mathbf{a}}}{\vec{\mathbf{a}}^t\mathbf{W}\vec{\mathbf{a}}}$$

This will be the first discriminant function.

Generalized eigenvalue problem

Theorem (Generalised eigenvalue problem)

Let $\mathbf{A}_{p\times p}$ and $\mathbf{B}_{p\times p}$ be symmetric matrices ($\mathbf{B}$ with only positive eigenvalues).

- Maximising the ratio
$$\frac{\vec{\mathbf{x}}^t\mathbf{A}\vec{\mathbf{x}}}{\vec{\mathbf{x}}^t\mathbf{B}\vec{\mathbf{x}}}$$
is associated with the first eigenpair of matrix $\mathbf{B}^{-1}\mathbf{A}$, $(\lambda_1, \vec{\mathbf{x}}_1)$.
- Sucessive pairs of eigenvalues/vectors of $\mathbf{B}^{-1}\mathbf{A}$ sucessively maximise the ratio $\frac{\vec{\mathbf{x}}^t\mathbf{A}\vec{\mathbf{x}}}{\vec{\mathbf{x}}^t\mathbf{B}\vec{\mathbf{x}}}$, subject to the $\mathbf{B}$-orthogonality of sucessive vectors, i.e., $\vec{\mathbf{x}}_i^t\mathbf{B}\vec{\mathbf{x}}_j = 0$, if $i \neq j$.

Note: The product of symmetric matrices is not, in general, symmetric, so their eigenvalues/vectors may be complex. But the eigenvalues/vectors of $\mathbf{B}^{-1}\mathbf{A}$ are necessarily real.

Fisher's formulation (cont.)

Solution: If **W** is invertible, the generalised eigenvalue problem (slide 94) gives the solution: take $\vec{\mathbf{a}} = \vec{\mathbf{a}}_1$, the eigenvector of $\mathbf{W}^{-1}\mathbf{B}$ with the largest eigenvalue.

The eigenvalue $\lambda_1 = \frac{\vec{\mathbf{a}}_1^t \mathbf{B} \vec{\mathbf{a}}_1}{\vec{\mathbf{a}}_1^t \mathbf{W} \vec{\mathbf{a}}_1}$ is the discriminating capacity of the axis: the ratio of bewteen-group and within-group variability

If the number of non-zero eigenvalues of $\mathbf{W}^{-1}\mathbf{B}$ is greater than 1, we may seek new discriminant linear combinations.

Sucessive solutions will be the linear combinations $\mathbf{X}\vec{\mathbf{a}}_j$ with $\vec{\mathbf{a}}_j$ given by other eigenvectors of matrix $\mathbf{W}^{-1}\mathbf{B}$ with non-zero eigenvalues.

The discriminating capacity of the new axes is given by their eigenvalues $\lambda_j = \frac{\vec{\mathbf{a}}_j^t \mathbf{B} \vec{\mathbf{a}}_j}{\vec{\mathbf{a}}_j^t \mathbf{W} \vec{\mathbf{a}}_j}$.

Observations

- If $k > n-p$, **W** is not invertible. In general, if $k \leq n-p$ **W** is invertible.
- The matrices of an LDA verify the relation $\mathbf{S} = \mathbf{B} + \mathbf{W}$.

$$\mathbf{I}_n = \mathbf{P}_G + (\mathbf{I}_n - \mathbf{P}_G) \quad \Rightarrow \quad \mathbf{X}^{c\,t}\mathbf{I}_n\mathbf{X}^c = \mathbf{X}^{c\,t}\mathbf{P}_G\mathbf{X}^c + \mathbf{X}^{c\,t}(\mathbf{I}_n - \mathbf{P}_G)\mathbf{X}^c \quad \Leftrightarrow \quad \mathbf{S} = \mathbf{B} + \mathbf{W}$$

- Sucessive discriminant axes are uncorrelated.

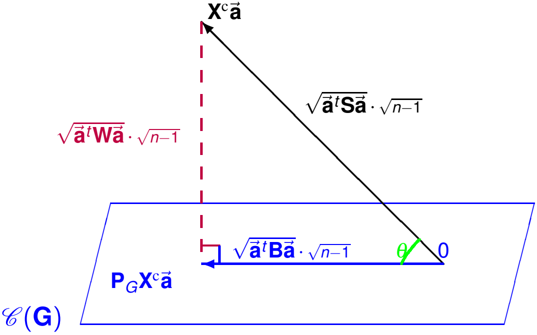
Different discriminant axes are of the form $\mathbf{X}^c\vec{\mathbf{a}}_i$ and $\mathbf{X}^c\vec{\mathbf{a}}_j$, with $\vec{\mathbf{a}}_i$ and $\vec{\mathbf{a}}_j$ different eigenvectors of matrix $\mathbf{W}^{-1}\mathbf{B}$. We know that $\vec{\mathbf{a}}_i$ and $\vec{\mathbf{a}}_j$ are **W**-orthogonal. Hence, if $i \neq j$:

$$\begin{aligned}
\mathbf{W}^{-1}\mathbf{B}\vec{\mathbf{a}}_j = \lambda_j\vec{\mathbf{a}}_j \quad & \Rightarrow \quad \mathbf{B}\vec{\mathbf{a}}_j = \lambda_j\mathbf{W}\vec{\mathbf{a}}_j \\
& \Rightarrow \quad \mathbf{W}\vec{\mathbf{a}}_j + \mathbf{B}\vec{\mathbf{a}}_j = \mathbf{W}\vec{\mathbf{a}}_j + \lambda_j\mathbf{W}\vec{\mathbf{a}}_j \\
& \Rightarrow \quad \mathbf{S}\vec{\mathbf{a}}_j = (1+\lambda_j)\mathbf{W}\vec{\mathbf{a}}_j \\
& \Rightarrow \quad Cov(\mathbf{X}^c\vec{\mathbf{a}}_i, \mathbf{X}^c\vec{\mathbf{a}}_j) = \vec{\mathbf{a}}_i^t\mathbf{S}\vec{\mathbf{a}}_j = (1+\lambda_j)\vec{\mathbf{a}}_i^t\mathbf{W}\vec{\mathbf{a}}_j = 0
\end{aligned}$$

Observations (cont.)

- Unlike PCA, discriminant axes $\mathbf{X}\vec{\mathbf{a}}_j$ being uncorrelated does not mean that the vectors of loadings $\vec{\mathbf{a}}_j$ are orthogonal (they are $\mathbf{W}$-orthogonal).
- $\mathbf{W}^{-1}\mathbf{B}$ cannot have more than $k-1$ non-zero eigenvalues. Thus, the number of discriminant axes cannot exceed the number of factor levels minus one.
- The solutions of a Linear Discriminant Analysis are invariant to linear transformations in the individual variables.

LDA - Summary



$\mathbf{X}^c\vec{\mathbf{a}}$
$\sqrt{\vec{\mathbf{a}}^t\mathbf{S}\vec{\mathbf{a}}}\cdot\sqrt{n-1}$
$\sqrt{\vec{\mathbf{a}}^t\mathbf{W}\vec{\mathbf{a}}}\cdot\sqrt{n-1}$
$\sqrt{\vec{\mathbf{a}}^t\mathbf{B}\vec{\mathbf{a}}}\cdot\sqrt{n-1}$
θ
0
$\mathbf{P}_G\mathbf{X}^c\vec{\mathbf{a}}$
$\mathscr{C}(\mathbf{G})$

Maximising $\frac{\vec{\mathbf{a}}^t\mathbf{B}\vec{\mathbf{a}}}{\vec{\mathbf{a}}^t\mathbf{W}\vec{\mathbf{a}}}$ means maximising $\mathrm{ctg}^2(\theta)$. For each axis, $\lambda_j = \mathrm{ctg}^2(\theta_j)$.

Again the geometry of LDA

Maximising the co-tangent of the angle θ means minimising θ.

In LDA we seek the linear combination $\mathbf{X}^c\vec{\mathbf{a}}$ of the centred variables (columns of $\mathbf{X}^c$) that form the smallest possible angle (θ) with the space spanned by the indicator variables of the subgroups (columns of $\mathbf{G}$).

This angle θ is the smallest angle between two subspaces of $\mathbb{R}^n$:

- the subspace spanned by the indicator variables, $\mathscr{C}(\mathbf{G})$; and
- the subspace spanned by the centred variables, $\mathscr{C}(\mathbf{X}^{\mathrm{c}})$.

The discriminant capacity of the variables depends on this smallest angle between $\mathscr{C}(\mathbf{X}^{\mathrm{c}})$ e $\mathscr{C}(\mathbf{G})$, i.e., on the angular relation between those two subspaces of $\mathbb{R}^n$.

Alternative formulations

Alternative formulations that minimise the angle θ:

1. Minimise the squared sine of θ.

   i.e., minimise the proportion of total variablity of the linear combination $\mathbf{X}^c\vec{\mathbf{a}}$ that corresponds to within-class variability.

$$\frac{\vec{\mathbf{a}}^t\mathbf{W}\vec{\mathbf{a}}}{\vec{\mathbf{a}}^t\mathbf{S}\vec{\mathbf{a}}}$$

2. Maximise the squared cosine of angle θ

   that is, maximise the proportion of total variability of the linear combination $\mathbf{X}^c\vec{\mathbf{a}}$ that corresponds to between-class variability.

$$\frac{\vec{\mathbf{a}}^t\mathbf{B}\vec{\mathbf{a}}}{\vec{\mathbf{a}}^t\mathbf{S}\vec{\mathbf{a}}}$$

Relations between alternative formulations

But the same problem (minimising θ) $\Rightarrow$ the same solution.

It is easy to check the equality of:

- Eigenvectors of $\mathbf{W}^{-1}\mathbf{B}$;
- Eigenvectors of $\mathbf{S}^{-1}\mathbf{W}$;
- Eigenvectors of $\mathbf{S}^{-1}\mathbf{B}$;

The linear combinations $\mathbf{X}^c\vec{\mathbf{a}}$ obtained with the alternative formulations are the same.

The corresponding eigenvalues are not equal because they correspond to different trigonometric functions. But they are related: let $\vec{\mathbf{a}}$ be the common eigenvector of all three matrices. Then:

- If λ is the corresponding eigenvector for matrix $\mathbf{W}^{-1}\mathbf{B}$;
- $\frac{1}{\lambda+1}$ is the eigenvector with matrix $\mathbf{S}^{-1}\mathbf{W}$ (which we minimise);
- $\frac{\lambda}{\lambda+1}$ is the eigenvalue with $\mathbf{S}^{-1}\mathbf{B}$ (which we maximise).

ADL e ANOVA

Consider:

- a one-way ANOVA with k factor levels (classes);
- the response variable $\vec{\mathbf{y}} = \mathbf{X}^c\vec{\mathbf{a}}$.

The criterion that defines LDA is equivalent to seeking $\vec{\mathbf{a}}$ such that the ANOVA F-test statistic for factor effects is maximum in:

- a one-way ANOVA with k factor levels (classes);
- with the response variable $\vec{\mathbf{y}} = \mathbf{X}^c\vec{\mathbf{a}}$.

The criterion that defines LDA is equivalent to seeking $\vec{\mathbf{a}}$ such that the ANOVA F-test statistic for factor effects is maximum:

$$F = \frac{QMF}{QMRE} = \frac{SQF}{SQRE} \cdot \frac{n-k}{k-1} = \frac{\|\mathbf{P}_G\vec{\mathbf{y}}\|^2}{\|(\mathbf{I}_n - \mathbf{P}_G)\vec{\mathbf{y}}\|^2} \cdot \frac{n-k}{k-1} = \frac{\vec{\mathbf{a}}^t\mathbf{B}\vec{\mathbf{a}}}{\vec{\mathbf{a}}^t\mathbf{W}\vec{\mathbf{a}}} \cdot \frac{n-k}{k-1} .$$

Discriminant axes are the successively uncorrelated linear combinations of the p observed variables that maximise the separation of values for each factor level.

Classification of new individuals using one axis

We can classify new individuals, of unknown “affiliation”.

Let $\vec{\mathbf{x}}$ be a vector of observations of the new individual on the p variables. The individual’s value (score) on the discriminant axis 1 is $y^* = \vec{\mathbf{x}}^t\vec{\mathbf{a}}_1$.

Comparing this value with the k class means on that axis, $\overline{y}^{(1)}, \overline{y}^{(2)}, \ldots, \overline{y}^{(k)}$, we can classify that individual in the group whose centre of gravity:

- is closest, in the usual Euclidean distance:

  attribute it to class i if $|y^* - \overline{y}^{(i)}| < |y^* - \overline{y}^{(j)}|, \quad \forall j \neq i.$

- is closest, on a Euclidean distance inversely weighted by the class standard deviation:

  attribute it to class i if $\frac{|y^* - \overline{y}^{(i)}|}{s_y^{(i)}} < \frac{|y^* - \overline{y}^{(j)}|}{s_y^{(j)}}, \quad \forall j \neq i,$

  where $s_y^{(i)}$ indicates the standard deviation of the scores in group i.

Classification with q axes

Using q discriminant axes, an individual has a vector of scores given by: $\vec{\mathbf{y}}^* = \vec{\mathbf{x}}^t \mathbf{A}_q$, with $\mathbf{A}_q$ the $p \times q$ matrix whose columns are the vectors $\vec{\mathbf{a}}_1, ..., \vec{\mathbf{a}}_q$.

We can classify the individual in the class whose centre of gravity $\vec{\mathbf{m}}_{(i)}$:

- is closest to $\vec{\mathbf{y}}^*$, in the usual Euclidean distance:
  attribute to class i if $\|\vec{\mathbf{y}}^* - \vec{\mathbf{m}}_{(i)}\| < \|\vec{\mathbf{y}}^* - \vec{\mathbf{m}}_{(j)}\|, \quad \forall j \neq i$

- is closest to $\vec{\mathbf{y}}^*$ in the usual Mahalanobis distance:
  attribute to class i if $\|\vec{\mathbf{y}}^* - \vec{\mathbf{m}}_{(i)}\|_{\mathbf{S}^{-1}} < \|\vec{\mathbf{y}}^* - \vec{\mathbf{m}}_{(j)}\|_{\mathbf{S}^{-1}}, \quad \forall j \neq i,$

  where $\mathbf{S}$ is the matrix of (co)variances of the scores of the n observations.

- is closest to $\vec{\mathbf{y}}^*$ in the Mahalanobis distances defined by the covariance matrix for the scores in each class:
  class i if $\|\vec{\mathbf{y}}^* - \vec{\mathbf{m}}_{(i)}\|_{\mathbf{S}_i^{-1}} < \|\vec{\mathbf{y}}^* - \vec{\mathbf{m}}_{(j)}\|_{\mathbf{S}_j^{-1}}, \forall j \neq i,$

  where $\mathbf{S}_i$ is the (co)variance matrix of the scores of group i.

LDA with `R` - command `lda`

Command `lda`, in the MASS package, provides the basic information for a Linear (Fisher) Discriminant Analysis.

The command `lda` was conceived for an inferential context (which is not ours and is not necessary for LDA). But it provides the essential information for a descriptive context.

Consider the example of the crayfish data: the first 21 observations are of reproducing males (group `MR`); the next 21 are non-reproducing males (group `MN`); and the final 21 observations are females (group `F`).

We create the factor of the groups and load the package `MASS`:

```
> lav.grupos <- factor(rep(c("MR","MN","F"),c(21,21,21)))
> library(MASS)
```

The crayfish data

LDA for the crayfish data lda

In the formula argument, the factor with the groups is the response variable.

```
> lav.lda <- lda(lav.grupos ~ . , data=as.data.frame(lavagantes))
> lav.lda
```

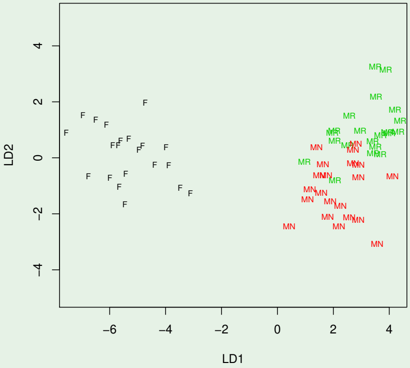
```
Coefficients of linear discriminants:                     <- loadings vectors
                      LD1          LD2
carapace_l    -0.0005163473 -1.19955746
tail_l        -0.1736612417  0.33191555
carapace_w     0.1866238904 -0.90101141
carapace_d    -0.3521185558 -0.23124418
tail_w        -2.6055856004  1.28663805
areola_l       0.3588957427 -0.06043209
areola_w      -2.1123185437 -0.03550332
rostrum_l      1.2415578489  1.22874815
rostrum_w     -0.3314912527  1.39715849
postorbital_w  0.1940959791 -1.59005854
propodus_l     0.6321803333  0.17783018
propodus_w     0.4297842346  0.71193763
dactyl_l      -0.0850563760  0.36615202

Proportion of trace:            <- proportion of the sum of non-zero eigenvalues of inv(W)B
   LD1    LD2
0.9501 0.0499               <-  not the criterion values defined above
```

Crayfish example (cont.)

The `plot` method for objects of class `lda` produced by the command `lda`:

Crayfish data LDA

```
> plot(lav.lda, col=as.numeric(lav.grupos))
```



Crayfish example (cont.)

The vectors of scores used to create the plot can be obtained with the command `predict`, in output argument `x`.

Crayfish LDA

```
> predict(lav.lda)$x

          LD1        LD2
1   2.9590031  0.9654792
2   3.6848954  0.8131683
3   3.5259200  2.1811447
4   2.0462745  0.6083346
[...]
60 -4.9547011  0.2934347
61 -6.7592582 -0.6571673
62 -5.6927267  0.4566755
63 -5.4276951 -0.5692571
```

The command `predict` can be used to determine the coordinates on the discriminant axes of a new individual, as was done in Linear Models.

Crayfish example (cont.)

Crayfish LDA

We define 3 new individuals whose values are the maximum values for each variable in each subgroup, and plot them on the new discriminant axes:

```
> lxm1 <- apply(lavagantes[1:21,],2,max)
> lxm2 <- apply(lavagantes[22:42,],2,max)
> lxm3 <- apply(lavagantes[43:63,],2,max)
> novos <- as.data.frame(rbind(lxm1,lxm2,lxm3))
> novos
```

```
     carapace_l tail_l carapace_w carapace_d tail_w areola_l areola_w rostrum_l
lxm1      35.33  25.15      18.36      14.57  15.40    13.26     2.60      7.06
lxm2      35.50  25.05      18.74      15.11  15.11    16.85     2.64      7.05
lxm3      35.73  26.77      18.50      15.06  17.37    13.14     2.32      7.27
     rostrum_w postorbital_w propodus_l propodus_w dactyl_l
lxm1      8.12         10.76      33.24      15.53    20.71
lxm2      7.74         11.85      33.67      15.15    20.83
lxm3      7.83         11.14      28.29      12.30    17.58
```

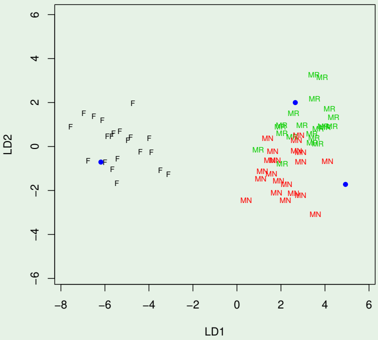
```
> predict(lav.lda, new=novos)$x
```

```
           LD1        LD2
lxm1  2.650187  1.9990716          <- coordinates of the first individual on the DAs
lxm2  4.931230 -1.7232999          <- coordinates of the second individual on the DAs
lxm3 -6.183037 -0.7078646          <- coordinates of the third individual on the DAs
```

Crayfish example (cont.)

Crayfish LDA

```
> lxmp <- predict(lav.lda, new=novos)$x
> plot(lav.lda, col=as.numeric(lav.grupos), xlim=c(-8,6))
> points(lxmp, col="blue", pch=16)
```



Remarks about the `lda` command in `MASS`

Attention: With the command `lda`,

- the **W** matrix is defined as $\mathbf{W} = \frac{1}{n-k}\mathbf{X}^{c\,t}(\mathbf{I}_n - \mathbf{P}_G)\mathbf{X}^c$;
- the **B** matrix is defined as $\mathbf{B} = \frac{1}{k-1}\mathbf{X}^{c\,t}\mathbf{P}_G\mathbf{X}^c$;
- the decomposition $\mathbf{S} = \mathbf{W} + \mathbf{B}$ no longer holds.
- the eigenvalues of $\mathbf{W}^{-1}\mathbf{B}$ with the `lda` definitions are $\frac{n-k}{k-1}$ times those of our definition. They are the value of the F-test statistic in the one-way ANOVA of each discriminant axis on the grouping factor;
- the `svd` component of an object of class `lda` gives the square roots of the eigenvalues of $\mathbf{W}^{-1}\mathbf{B}$ (defined as in `lda`).

The quality of the discriminant axes

Crayfish LDA

```
> lav.lda$svd
[1] 21.345129  4.890076
> lav.lda$svd^2  <- Eigenvalues (and values of the F statistic)
[1] 455.61455  23.91285

> summary(aov(predict(lav.lda)$x[,1] ~ lav.grupos))
            Df Sum Sq Mean Sq F value Pr(>F)
lav.grupos   2  911.2   455.6   455.6 <2e-16
Residuals   60   60.0     1.0

> summary(aov(predict(lav.lda)$x[,2] ~ lav.grupos))
            Df Sum Sq Mean Sq F value   Pr(>F)
lav.grupos   2  47.83   23.91   23.91 2.31e-08
Residuals   60  60.00    1.00
```

Quality of discriminant axes (cont.)

The eigenvalues of Fisher's original definition are given by multiplying the lda eigenvalues by $\frac{k-1}{n-k}$.

Crayfish LDA

```
> lav.lda$svd^2*2/60  <- Eigenvalues with Fisher's definition of W and B
[1] 15.1871516  0.7970949
```

The discriminating capacity of the first axis is 15.187, i.e., the variability between the three groups is, on that axis, 15.187 times larger than the within-group variability.

The discriminating capacity of the second axis is weak: 0.797, i.e., on that axis, the variability between the three groups is smaller than the variability within groups.

Remarks about the `lda` function (cont.)

- the proportions of the trace (given in the output) of each eigenvalue are not affected by the different definitions.

```
> val <- lav.lda$svd^2
> val/sum(val)
[1] 0.95013247 0.04986753
> val2 <- lav.lda$svd^2*2/60
> val2/sum(val2)
[1] 0.95013247 0.04986753
```

- the **W**-orthogonality of the loadings given in the output is also preserved (although the squared norm of the loadings vectors is affected: it is $\frac{n-k}{n-1}$ when measured using the definition of **W** on slide 93).

The classification of new individuals

The predict method of the lda command classifies individuals in the groups, with criteria based on inferential concepts, but analogous to classifications based on Mahalanobis distances. The classifications are stored in the class output object.

Classification of crayfish with LDA

```
> predict(lav.lda)$class

[1] MR MR MR MR MR MR MR MN MN MR MR MR MR MR MR MR MR MR MR MR MR MN MR MN MN
[26] MN MR MN MN MN MN MN MN MN MN MN MN MN MR MN MN MN F  F  F  F  F  F  F  F
[51] F  F  F  F  F  F  F  F  F  F  F  F  F

> predict(lav.lda, new=novos)$class

[1] MR MN F
```

Classification tables

Classification tables may be created with the `table` command.

Classification tables for the crayfish data

```
> lav.pred <- predict(lav.lda)$class
> table(lav.pred, lav.grupos)
        lav.grupos
lav.pred  F MN MR
      F  21  0  0
      MN  0 18  2
      MR  0  3 19
```

- All the females were correctly classified.
- Three non-reproducing males were incorrectly classified as reproducing males.
- Two reproducing males were incorrectly classified as non-reproducing males.

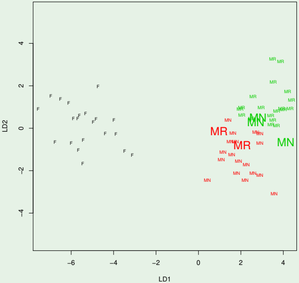
Misclassifications

Wrong classifications in the crayfish LDA

```
> (lav.grupos != predict(lav.lda)$class)
```

```
 [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[17] FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
[33] FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[49] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
> lav.mal <- (lav.grupos != predict(lav.lda)$class)
> plot(lav.lda, col=as.numeric(predict(lav.lda)$class), cex=0.7+lav.mal)
```



R's potential: a new function for LDA

Exercise 17: our function adl

```
> adl <- function(X,grupos){      <- input arguments are data (X) and classes (grupos)
grupos <- as.factor(grupos)       <- ensures that 'grupos' is a factor
X <- as.matrix(X)                 <- ensures that 'X' is a matrix
k <- length(levels(grupos))       <- k: number of groups
n <- dim(X)[1]                    <- n: number of individuals
p <- dim(X)[2]                    <- p: number of variables
Ind <- model.matrix(aov(X[,1] ~ -1 + grupos))  <- creates matrix G as in the slides
PG <- Ind %*% solve(t(Ind)%*%Ind) %*% t(Ind)   <- projection matrix P_G
Xc <- scale(X, scale=F)           <- centred data matrix
B <- (t(Xc) %*% PG %*% Xc)/(n-1)  <- between-group variability matrix B
W <- (t(Xc) %*% (diag(n)-PG) %*% Xc)/(n-1)      <- within-group variability matrix W
valvec <- eigen(solve(W)%*%B)     <- eigenvalues and eigenvectors of inv(W)B
val <- Re(valvec$val)[1:(k-1)]    <- eigenvalues of inv(W)B
loadings <- Re(valvec$vec)[,1:(k-1)]   <- eigenvectors of inv(W)B
if (k>2) rownames(loadings) <- colnames(X)   <- names of objects in the output list
else if (k==2) names(loadings) <- colnames(X)
rownames(B) <- colnames(X)
colnames(B) <- colnames(X)
rownames(W) <- colnames(X)
colnames(W) <- colnames(X)
if (k>2) colnames(loadings) <- paste("ED",1:(k-1),sep=)
scores <- Xc %*% loadings
rownames(scores) <- rownames(X)
list(B=B,W=W,val=val,loadings=loadings,scores=scores)  <- output object (list)
    }
```

The adl function in action

```
> adl(lavagantes, lav.grupos)$val
[1] 15.1871516  0.7970949       <- compare with previous values

> adl(lavagantes, lav.grupos)$loadings      <- of norm 1, W-orthogonal

                     ED1         ED2
carapace_l     0.000138748 -0.36607521
tail_l         0.046664632  0.10129240
carapace_w    -0.050147834 -0.27496636
carapace_d     0.094618019 -0.07056999
tail_w         0.700148699  0.39265005
areola_l      -0.096439122 -0.01844238
areola_w       0.567602569 -0.01083473
rostrum_l     -0.333619864  0.37498349
rostrum_w      0.089075243  0.42637815
postorbital_w -0.052155664 -0.48524647
propodus_l    -0.169873612  0.05426936
propodus_w    -0.115487617  0.21726572
dactyl_l       0.022855557  0.11174052
```