Certificate Data Science for Managment

Introduction to Dimensionality Reduction

X – HEC, Spring 2020

Julien Chiquet

https://jchiquet.github.io/ds4m

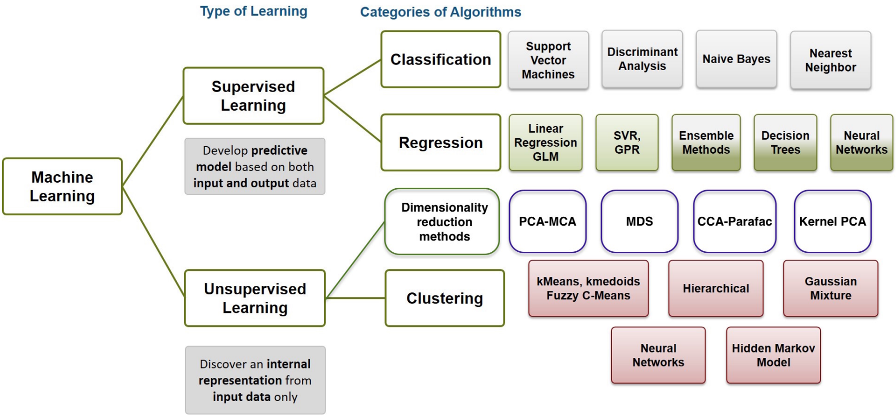
Part I

Introduction

Packages required for reproducing the slides

```
library(tidyverse)  # opinionated collection of packages for data manipulation
library(GGally)     # extension to ggplot vizualization system
library(FactoMineR) # PCA and oter linear method for dimension reduction
library(factoextra) # fancy plotting for FactoMineR output
# color and plots themes
library(RColorBrewer)
pal <- brewer.pal(10, "Set3")
theme_set(theme_bw())
```

Machine Learning



Supervised vs Unsupervised Learning

Supervised Learning

- Training data $\mathcal{D}_n = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}, X_i \sim^{\text{i.i.d}} \mathbb{P}$
- Construct a predictor $\hat{f} : \mathcal{X} \to \mathcal{Y}$ using $\mathcal{D}_n$
- Loss $\ell(y, f(x))$ measures how well $f(x)$ predicts y
- Aim: minimize the generalization error
- Task: Regression, Classification

$\rightsquigarrow$ The goal is clear: predict y based on x (regression, classification)

Unsupervised Learning

- Training data $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$
- Loss? , Aim?
- Task: **Dimension reduction**, Clustering

$\rightsquigarrow$ The goal is less well defined, and *validation* is questionable

Dimension Reduction?

Figure: source: F. Belardi

- How to view a high-dimensional dataset ?
- High-dimension: dimension larger than 2!
- *Projection* in a 2D space.

Dimension Reduction?

Figure: source: F. Belardi

- How to view a high-dimensional dataset ?
- High-dimension: dimension larger than 2!
- *Projection* in a 2D space.

Dimension Reduction?

Figure: source: F. Belardi

- How to view a high-dimensional dataset ?
- High-dimension: dimension larger than 2!
- *Projection* in a 2D space.

Companion data set: 'crabs'

Morphological Measurements on Leptograpsus Crabs

Description: *small data, low-dimensional*

The crabs data frame has 200 rows and 8 columns, describing 5 morphological measurements on 50 crabs each of two colour forms and both sexes, of the species *Leptograpsus variegatus* collected at Fremantle, W. Australia.

Figure: A leptograpsus Crab

Companion data set: 'crabs' I

Table header

```r
crabs <- MASS::crabs %>% select(-index) %>%
  rename(sex = sex,
         species          = sp,
         frontal_lob      = FL,
         rear_width       = RW,
         carapace_length = CL,
         carapace_width  = CW,
         body_depth       = BD)
crabs %>% select(sex, species) %>% summary() %>% knitr::kable("latex")
```

	sex	species
	F:100	B:100
	M:100	O:100

```r
dim(crabs)

## [1] 200   7
```

Companion data set: 'crabs' II

Table header

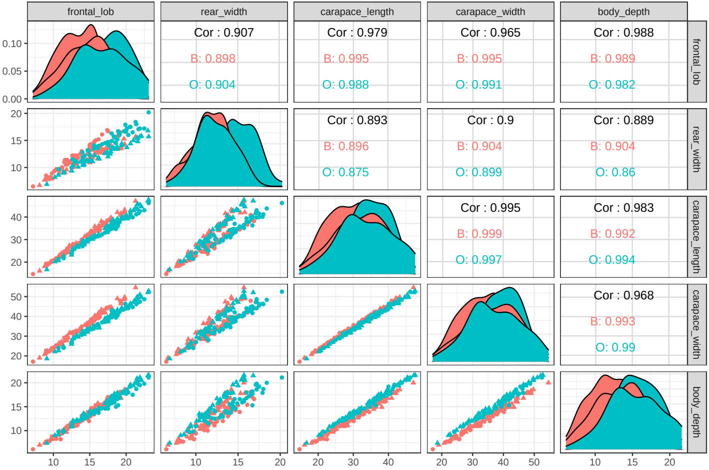
```r
crabs %>% head(15) %>% knitr::kable("latex")
```

species	sex	frontal_lob	rear_width	carapace_length	carapace_width	body_depth
B	M	8.1	6.7	16.1	19.0	7.0
B	M	8.8	7.7	18.1	20.8	7.4
B	M	9.2	7.8	19.0	22.4	7.7
B	M	9.6	7.9	20.1	23.1	8.2
B	M	9.8	8.0	20.3	23.0	8.2
B	M	10.8	9.0	23.0	26.5	9.8
B	M	11.1	9.9	23.8	27.1	9.8
B	M	11.6	9.1	24.5	28.4	10.4
B	M	11.8	9.6	24.2	27.8	9.7
B	M	11.8	10.5	25.2	29.3	10.3
B	M	12.2	10.8	27.3	31.6	10.9
B	M	12.3	11.0	26.8	31.5	11.4
B	M	12.6	10.0	27.7	31.7	11.4
B	M	12.8	10.2	27.2	31.8	10.9
B	M	12.8	10.9	27.4	31.5	11.0

Companion data set: 'crabs'

Pairs plot of attributes

```
ggpairs(crabs, columns = 3:7, aes(colour = species, shape = sex))
```



⇝ Pairs plot don't help...

Companion data set: 'crabs'

Correlation matrix

```
crabs %>%  select(-species, -sex) %>% cor( ) %>% kable('latex', digits = 3)
```

	frontal_lob	rear_width	carapace_length	carapace_width	body_depth
frontal_lob	1.000	0.907	0.979	0.965	0.988
rear_width	0.907	1.000	0.893	0.900	0.889
carapace_length	0.979	0.893	1.000	0.995	0.983
carapace_width	0.965	0.900	0.995	1.000	0.968
body_depth	0.988	0.889	0.983	0.968	1.000

Very high correlation!

- much redundancy?
- hidden factor?

$\rightsquigarrow$ dimension reduction might hem

Another example: 'snp'

Genetics variant in European population

Description: *medium/large data, high-dimensional*

500, 000 Genetics variants (SNP – Single Nucleotide Polymorphism) for 3000 individuals (1 meter $\times$ 166 meter (height $\times$ width)

- SNP : 90 % of human genetic variations
- coded as 0, 1 or 2 (10, 1 or 2 allel different against the population reference)

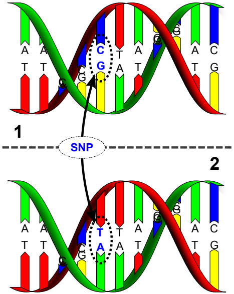


Figure: SNP (wikipedia)

Summarize 500,000 variables in 2

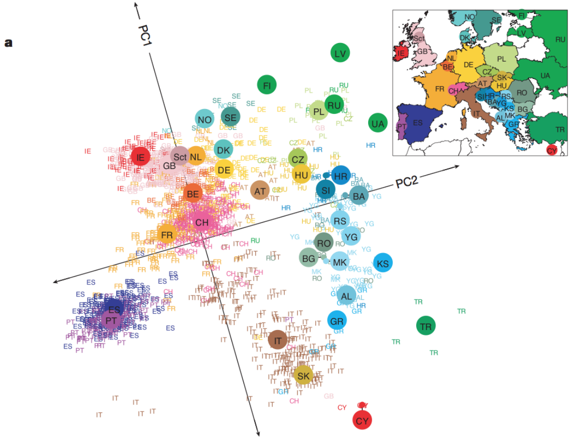


Figure: PCA output source: Nature "Gene Mirror Geography Within Europe", 2008

⇝ How much information is lost?

Theoretical argument: dimensionality Curse

High Dimension Geometry Curse

- Folks theorem: In high dimension, everyone is alone.
- Theorem: If $\mathbf{x}_1, \ldots, \mathbf{x}_n$ in the hypercube of dimension d such that their coordinates are i.i.d then

$$d^{-1/p}\left(\max \|\mathbf{x}_i - \mathbf{x}_{i'}\|_p - \min \|\mathbf{x}_i - \mathbf{x}_{i'}\|_p\right) = 0 + O\left(\sqrt{\frac{\log n}{d}}\right)$$

$$\frac{\max \|\mathbf{x}_i - \mathbf{x}_{i'}\|_p}{\min \|\mathbf{x}_i - \mathbf{x}_{i'}\|_p} = 1 + O\left(\sqrt{\frac{\log n}{d}}\right).$$

$\rightsquigarrow$ When d is large, all the points are almost equidistant
Hopefully, the data **are not really leaving in d** dimension (think of the SNP example)

Dimension reduction: goals summary

Main objective: find a **low-dimensional representation** that captures the "essence" of (high-dimensional) data

Application in Machine Learning

Preprocessing, Regularization

- compression, denoising, anomaly detection
- Reduce overfitting in supervised learning (improve performances)

Application in statistics and data analysis

Better understand the data

- descriptive/exploratory methods
- visualization: difficult to plot and interpret $> 3d$!

Dimension reduction: problem setup

Settings

- Training data : $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \in \mathbb{R}^d$, (i.i.d.)
- Space $\mathbb{R}^d$ of possibly high dimension $(n \ll d)$

Dimension Reduction Map

Construct a map Φ from the space $\mathbb{R}^d$ into a space $\mathbb{R}^{d'}$ of smaller dimension:

$$\begin{aligned} \Phi : \quad & \mathbb{R}^d \to \mathbb{R}^{d'}, d' \ll d \\ & \mathbf{x} \mapsto \Phi(\mathbf{x}) \end{aligned}$$

How should we design/construct Φ?

Criterion

- **Geometrical approach**
- Reconstruction error
- Relationship preservation

Form of the map Φ

- **Linear** or non-linear ?
- tradeoff between **interpretability** and versatility ?
- tradeoff between high or **low** computational resource

Part II

Principal Component Analysis

Some references...

...biased choices!

- Analyse en composantes principales, Course AgroParisTech
  Carine Ruby, Stéphane Robin
  http://www.agroparistech.fr/IMG/pdf/AnalyseComposantesPrincipales-AgroParisTech.pdf
- Exploratory Multivariate Analysis by Example using R,
  Husson, Le, Pages, 2017.
  Chapman & Hall
- Multiple Factor Analysis by Example using R,
  J. Pagès 2015.
  CRC Press
- An Introduction to Statistical Learning
  G. James, D. Witten, T. Hastie and R. Tibshirani
  http://faculty.marshall.usc.edu/gareth-james/ISL/

PCA and classical Linear methods

Principal component Analysis (PCA) is for continuous data

Non continuous data

- Correspondence analysis (CA): contingency table
- Multiple correspondence analysis (MCA): categorical data
- Multiple factor analysis (MFA): multi-table, array data

$\rightsquigarrow$ Basic adaptation that build on PCA to deal with non-continuous data
$\rightsquigarrow$ smart encoding of non-continuous data to continuous ones

We will focus on PCA, as the mother or most linear (and non-linear) methods.

The data matrix

The data set is a $n \times d$ matrix $\mathbf{X} = (x_{ij})$ with values in $\mathbb{R}$:

- each row $\mathbf{x}_i$ represents an individual/observation
- each col $\mathbf{x}^j$ represents a variable/attribute

```r
crabs %>% head(6) %>% knitr::kable("latex")
```

species	sex	frontal_lob	rear_width	carapace_length	carapace_width	body_depth
B	M	8.1	6.7	16.1	19.0	7.0
B	M	8.8	7.7	18.1	20.8	7.4
B	M	9.2	7.8	19.0	22.4	7.7
B	M	9.6	7.9	20.1	23.1	8.2
B	M	9.8	8.0	20.3	23.0	8.2
B	M	10.8	9.0	23.0	26.5	9.8

Objectives

Individual/Observations

- similarity between observations with respect to all the variables
- Find pattern ($\sim$ partition) between individuals

Variables

- linear relationships between variables
- visualization of the correlation matrix
- find synthetic variables

Link between the two

- characterization of the groups of individuals with variables
- specific observations to understand links between variables

Outline

Principal Component Analysis

1. Background: high-school algebra
2. Geometric approach to PCA
3. Principal axes and variance maximization
4. Representation and interpretation
5. Additional tools and Complements
6. Beyond linear methods

Vectors in $\mathbb{R}^n$

Definition and Basics

A vector $\mathbf{x} \in \mathbb{R}^d$ is defined by a d-uplet $(x_1, x_2, \ldots, x_d)$, *its coordinates*.

Elementary operations

- Addition of two vectors (define a parallelogram)

$$\mathbf{x} + \mathbf{y} = \begin{pmatrix} x_1 + y_1 \\ x_2 + y_2 \\ \vdots \\ x_d + y_d \end{pmatrix}$$

- Multiplication by a scalar (streching)

$$\lambda \mathbf{x} = \begin{pmatrix} \lambda x_1 \\ \lambda x_2 + c \\ \vdots \\ \lambda x_d \end{pmatrix}, \quad \lambda, c \in \mathbb{R}.$$

Properties

- associativity: $(\mathbf{x} + \mathbf{y}) + \mathbf{z} = \mathbf{x} + (\mathbf{y} + \mathbf{z})$
- commutativity: $\mathbf{x} + \mathbf{y} = \mathbf{y} + \mathbf{x}$
- linearity: $\lambda(\mathbf{x} + \mathbf{y}) = \lambda \mathbf{x} + \lambda \mathbf{y}$
- $(\lambda_1 + \lambda_2)\mathbf{x} = \lambda_1 \mathbf{x} + \lambda_2 \mathbf{x}$

Vectors in $\mathbb{R}^n$

Dot/Inner product and norm

Dot product of 2 vectors: sum of the products between each coordinate:

$$\langle \mathbf{x}, \mathbf{y} \rangle \equiv \mathbf{x} \cdot \mathbf{y} \equiv \mathbf{x}^\top \mathbf{y} \triangleq \sum_{i=1}^{d} x_i \, y_j .$$

- $\mathbf{x}^\top \mathbf{y} = \mathbf{y}^\top \mathbf{x}$
- $\mathbf{x}^\top (\mathbf{y} + \mathbf{z}) = \mathbf{x}^\top \mathbf{y} + \mathbf{x}^\top \mathbf{z}$
- $\lambda(\mathbf{x}^\top \mathbf{y}) = (\lambda(\mathbf{x})^\top \mathbf{y} = \mathbf{x}^\top (\lambda \mathbf{y})$
- if $\mathbf{x} = \mathbf{0}$, then $\mathbf{x}^\top \mathbf{x} = 0$.

(Euclidean) norm (a.k.a length, magnitude)

$$\|\mathbf{x}\| = \sqrt{\mathbf{x}^\top \mathbf{x}}. \quad \text{we have } \|\lambda \mathbf{x}\| = |\lambda| \|\mathbf{x}\|.$$

(Euclidean) distance between 2 vectors

$$\text{dist}(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|.$$

Remark that when $\mathbf{x}$ and $\mathbf{y}$ are orthogonal and non zero, distances between $\mathbf{x}$ and $\mathbf{y}$ and $\mathbf{x}$ and $(-\mathbf{y})$ are the same. Then,

$$(\mathbf{x} - \mathbf{y})^\top (\mathbf{x} - \mathbf{y}) = (\mathbf{x} + \mathbf{y})^\top (\mathbf{x} + \mathbf{y}) \Leftrightarrow \mathbf{x}^\top \mathbf{y} = 0,$$

which motivates the following definition of orthornality:

Orthogonality

Two vectors $\mathbf{x}, \mathbf{y} \neq \mathbf{0}$ are orthogonal iff $\mathbf{x}^\top \mathbf{y} = 0$.

Vectors in $\mathbb{R}^n$

Distances and orthogonality

(Euclidean) distance between 2 vectors

$$\text{dist}(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|.$$

Remark that when $\mathbf{x}$ and $\mathbf{y}$ are orthogonal and non zero, distances between $\mathbf{x}$ and $\mathbf{y}$ and $\mathbf{x}$ and $(-\mathbf{y})$ are the same. Then,

$$(\mathbf{x} - \mathbf{y})^\top (\mathbf{x} - \mathbf{y}) = (\mathbf{x} + \mathbf{y})^\top (\mathbf{x} + \mathbf{y}) \Leftrightarrow \mathbf{x}^\top \mathbf{y} = 0,$$

which motivates the following definition of orthornality:

Orthogonality

Two vectors $\mathbf{x}, \mathbf{y} \neq \mathbf{0}$ are orthogonal iff $\mathbf{x}^\top \mathbf{y} = 0$.

Vectors in $\mathbb{R}^n$

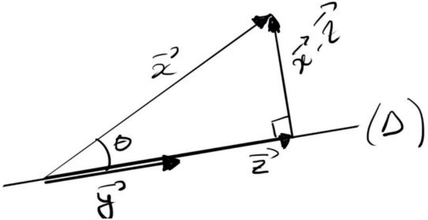
Orthogonal Projection and geometric definition of the dot product

Orthogonal projection of $\mathbf{x}$ onto $\mathbf{y}$

It is the vector $\mathbf{z}$ such that

1. $\mathbf{z} = \lambda \mathbf{y}$
2. $\mathbf{y}$ is orthogonal to $\mathbf{x} - \mathbf{z}$

We find $\lambda = \mathbf{x}^\top \mathbf{y} / \|\mathbf{y}\|^2$



Thanks to Pythagoras theorem,

$$\cos(\theta) = \frac{\|\mathbf{z}\|}{\|\mathbf{x}\|} = \lambda \frac{\|\mathbf{y}\|}{\|\mathbf{x}\|}$$

and then we end with the following geometric definition of the dot product

Dot product: geometric definition

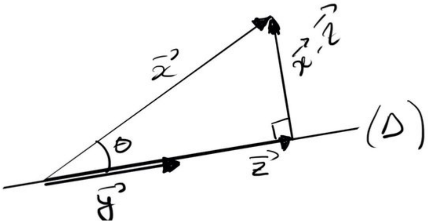
$$\mathbf{x}^\top \mathbf{y} = \cos(\theta) \|\mathbf{x}\| \, \|\mathbf{y}\|$$

Orthogonal projection of $\mathbf{x}$ onto $\mathbf{y}$

It is the vector $\mathbf{z}$ such that

1. $\mathbf{z} = \lambda \mathbf{y}$
2. $\mathbf{y}$ is orthogonal to $\mathbf{x} - \mathbf{z}$

We find $\lambda = \mathbf{x}^\top \mathbf{y} / \|\mathbf{y}\|^2$



Thanks to Pythagoras theorem,

$$\cos(\theta) = \frac{\|\mathbf{z}\|}{\|\mathbf{x}\|} = \lambda \frac{\|\mathbf{y}\|}{\|\mathbf{x}\|}$$

and then we end with the following geometric definition of the dot product

Dot product: geometric definition

$$\mathbf{x}^\top \mathbf{y} = \cos(\theta) \|\mathbf{x}\| \, \|\mathbf{y}\|$$

Outline

Principal Component Analysis

The data matrix

The data set is a $n \times d$ matrix $\mathbf{X} = (x_{ij})$ with values in $\mathbb{R}$:

- each row $\mathbf{x}_i$ represents an individual/observation
- each col $\mathbf{x}^j$ represents a variable/attribute

$$\mathbf{X} = \begin{array}{cc} & \begin{array}{cccccc} \mathbf{x}^1 & \mathbf{x}^2 & \dots & \mathbf{x}^j & \dots & \mathbf{x}^d \end{array} \\ \begin{array}{c} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_i \\ \vdots \\ \mathbf{x}_n \end{array} & \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1j} & \dots & x_{1d} \\ x_{21} & x_{22} & \dots & x_{2j} & \dots & x_{2d} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{i1} & x_{i2} & \dots x_{ij} & \dots & x_{id} & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{n1} & x_{n2} & \dots x_{nj} & \dots & x_{nd} & \end{pmatrix} \end{array}$$

```
crabs %>% head(3) %>% knitr::kable("latex")
```

species	sex	frontal_lob	rear_width	carapace_length	carapace_width	body_depth
B	M	8.1	6.7	16.1	19.0	7.0
B	M	8.8	7.7	18.1	20.8	7.4
B	M	9.2	7.8	19.0	22.4	7.7

Cloud of observation in $\mathbb{R}^d$

Individuals can be represented in the variable space $\mathbb{R}^d$ as a point cloud

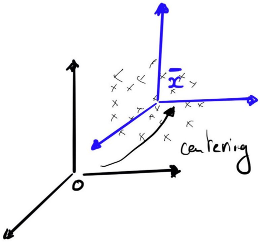


Figure: Example in $\mathbb{R}^3$

Center of Inertia

(or barycentrum, or empirical mean)

$$\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i = \begin{pmatrix} \sum_{i=1}^{n} x_{i1}/n \\ \sum_{i=1}^{n} x_{i2}/n \\ \vdots \\ \sum_{i=1}^{n} x_{id}/n \end{pmatrix}$$

We center the cloud $\mathbf{X}$ around $\mathbf{x}$ denote this by $\mathbf{X}^c$

$$\mathbf{X}^c = \begin{pmatrix} x_{11}-\bar{x}_1 & \dots & x_{1j}-\bar{x}_j & \dots & x_{1d}-\bar{x}_d \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{i1}-\bar{x}_1 & \dots & x_{ij}-\bar{x}_j & \dots & x_{id}-\bar{x}_d \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{n1}-\bar{x}_1 & \dots & x_{nj}-\bar{x}_j & \dots & x_{nd}-\bar{x}_d \end{pmatrix}$$

Inertia and Variance

Total Inertia: distance of the individuals to the center of the cloud

$$I_T = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{d}(x_{ij}-\bar{x}_j)^2 = \frac{1}{n}\sum_{i=1}^{n}\|\mathbf{x}_i-\bar{\mathbf{x}}\|^2 = \frac{1}{n}\sum_{i=1}^{n}\mathsf{dist}^2(\mathbf{x}_i,\bar{\mathbf{x}})$$

I_T is proportional to the total variance

Let $\hat{\boldsymbol{\Sigma}}$ be the empirical variance-covariance matrix

$$I_T = \frac{1}{n}\sum_{j=1}^{p}\sum_{i=1}^{n}(x_{ij}-\bar{x}_j)^2 = \sum_{j=1}^{n}\frac{1}{n}\|\mathbf{x}^j-\bar{x}_j\|^2 = \sum_{j=1}^{n}\mathbb{V}(\mathbf{x}^j) = \mathrm{trace}\left(\hat{\boldsymbol{\Sigma}}\right)$$

- $\rightsquigarrow$ Good representation has large inertia (much variability)
- $\rightsquigarrow$ Large dispertion $\sim$ Large distances between points

Inertia with respect to an axix

The Inertia of the cloud wrt axe Δ is the sum of the distances between all points and their orthogonal projection on Δ.

$$I_{\Delta} = \frac{1}{n} \sum_{i=1}^{n} \text{dist}^2(\mathbf{x}_i, \Delta)$$

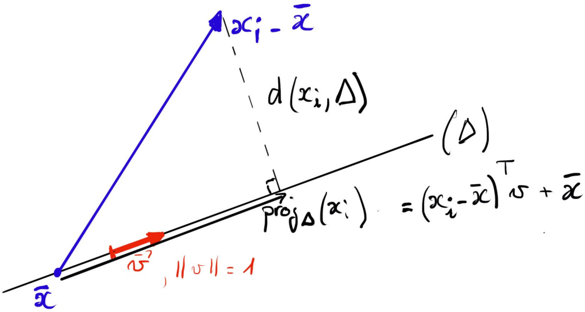
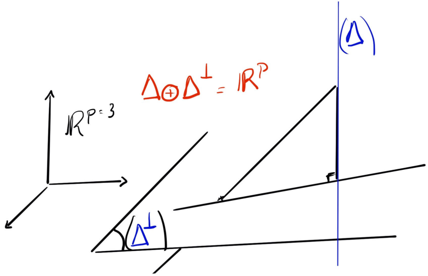


Figure: Projection of $\mathbf{x}_i$ onto a line Δ passing through $\bar{\mathbf{x}}$

Decomposition of total Inertia (1)

Let $\Delta^{\perp}$ the orthogonal subspace Δ is $\mathbb{R}^n$



Theorem (Huygens)

A consequence of the above (Pythagoras Theorem) is the decomposition of the following total inertia:

$$I_T = I_{\Delta} + I_{\Delta^{\perp}}$$

By projecting the cloud $\mathbf{X}$ onto Δ, with loss the inertia measured by $\Delta^{\perp}$

Decomposition of total Inertia (2)

Consider only subspaces with dimension 1 (that is, lines or axes). We can decompose $\mathbb{R}^p$ as the sum of p othogonal axis.

$$\mathbb{R}^p = \Delta_1 \oplus \Delta_2 \oplus \cdots \oplus \Delta_p$$

$\rightsquigarrow$ These axes form a new basis for representing the point cloud.

Theorem (Huygens)

$$I_T = I_{\Delta_1} + I_{\Delta_2} + \cdots + I_{\Delta_p}$$

Outline

Principal Component Analysis

1. Background: high-school algebra
2. Geometric approach to PCA
3. Principal axes and variance maximization
4. Representation and interpretation
5. Additional tools and Complements
6. Beyond linear methods

Finding the best axis (1)

Definition of the problem

- The best axis Δ_1 is the "closest" to the point cloud
- Inertia of Δ_1 measures the distance between the data and Δ_1
- Δ_1 is defined by the director vector $\mathbf{u}_1$, such as $\|\mathbf{u}_1\| = 1$
- $\Delta_1^\perp$ is defined by the normal vector $\mathbf{u}_1$, such as $\|\mathbf{u}_1\| = 1$

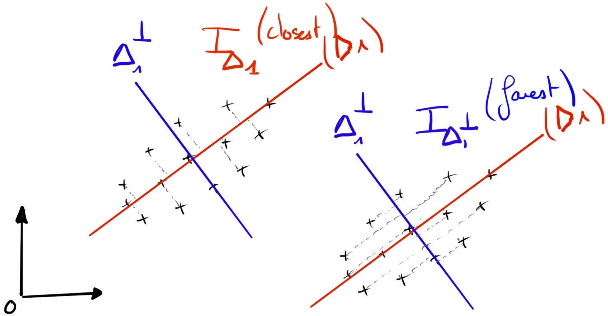
$\rightsquigarrow$ The best axis Δ_1 is the one with the minimal Inertia.

Finding the best axis (2)

Stating the optimization problem

Since $\Delta_1 \oplus \Delta_1^{\perp} = \mathbb{R}^p$ and $I_T = I_{\Delta_1} + I_{\Delta_1^{\perp}}$, then

$$\underset{\mathbf{u} \in \mathbb{R}^p : \|\mathbf{u}\| = 1}{\text{minimize}} \; I_{\Delta_1} \Leftrightarrow \underset{\mathbf{u} \in \mathbb{R}^p : \|\mathbf{u}\| = 1}{\text{maximize}} \; I_{\Delta_1^{\perp}}$$



Finding the best axis (3)

Stating the problem (algebraically)

Find $\mathbf{u}_1; \|\mathbf{u}_1\| = 1$ that minimizes

$$
\begin{aligned}
I_{\Delta_1^\perp} &= \frac{1}{n}\sum_{i=1}^{n} \mathrm{dist}(\mathbf{x}_i, \Delta_1^\perp)^2 \\
&= \frac{1}{n}\sum_{i=1}^{n} \mathbf{u}_1^\top(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\top \mathbf{u}_1 \\
&= \mathbf{u}_1^\top \left(\sum_{i=1}^{n} \frac{1}{n}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\top \right) \mathbf{u}_1 \\
&= \mathbf{u}_1^\top \hat{\boldsymbol{\Sigma}} \mathbf{u}_1
\end{aligned}
$$

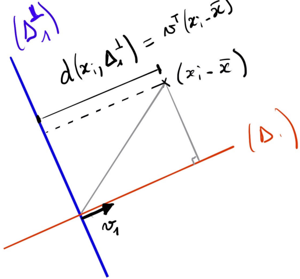


Figure: Geometrical insight

Finding the best axis (4)

We solve a simple constraint maximization problem with the method of Lagrange multipliers:

$$\underset{\mathbf{u}_1 : \|\mathbf{u}_1\|=1}{\text{maximize}} \, \mathbf{u}_1^\top \hat{\boldsymbol{\Sigma}} \mathbf{u}_1 \Leftrightarrow \underset{\mathbf{u}_1 \in \mathbb{R}^p, \lambda_1 > 0}{\text{maximize}} \, \mathbf{u}_1^\top \hat{\boldsymbol{\Sigma}} \mathbf{u}_1 - \lambda_1 (\|\mathbf{u}_1\| - 1)$$

By straightforward (vector) differentiation, an using that $\mathbf{u}_1^\top \mathbf{u}_1 = 1$

$$\begin{cases} 2\hat{\boldsymbol{\Sigma}}\mathbf{u}_1 - 2\lambda_1 \mathbf{u}_1 = 0 \\ \mathbf{u}_1^\top \mathbf{u}_1 - 1 = 0 \end{cases} \Leftrightarrow \begin{cases} \hat{\boldsymbol{\Sigma}}\mathbf{u}_1 = \lambda_1 \mathbf{u}_1 \\ \mathbf{u}_1^\top \hat{\boldsymbol{\Sigma}} \mathbf{u}_1 = \lambda_1 \mathbf{u}_1^\top \mathbf{u}_1 = \lambda_1 = I_{\Delta_1}^\perp \end{cases}$$

- $\mathbf{u}_1$ is the first eigen vector of $\hat{\boldsymbol{\Sigma}}$
- λ_1 is the first eigen value of $\hat{\boldsymbol{\Sigma}}$

$\rightsquigarrow$ **Δ_1 is defined by the first eigen vector of $\hat{\boldsymbol{\Sigma}}$**
$\rightsquigarrow$ **Variance "carried" by Δ_1 is equal to the largest eigen value of $\hat{\boldsymbol{\Sigma}}$**

Finding the following axes

Second best axis

Find Δ_2 with dimension 1, director vector $\mathbf{u}_2$ orthogonal to Δ_1 solving

$$\underset{\mathbf{u}_2 \in \mathbb{R}^p}{\text{maximize}}\, I_{\Delta_2^\perp} = \mathbf{u}_2^\top \hat{\boldsymbol{\Sigma}} \mathbf{u}_2, \quad \text{with } \|\mathbf{u}_2\| = 1, \mathbf{u}_1^\top \mathbf{u}_2 = 0.$$

$\rightsquigarrow$ $\mathbf{u}_2$ is the second eigen vector of $\hat{\boldsymbol{\Sigma}}$ with eigen value λ_2

And so on!

PCA is roughly a matrix factorisation problem

$$\hat{\boldsymbol{\Sigma}} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top, \quad \mathbf{U} = \begin{pmatrix} \mathbf{u}_1 & \mathbf{u}_2, & \dots & \mathbf{u}_p \end{pmatrix}, \quad \boldsymbol{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_p)$$

- $\mathbf{U}$ is an orthogonal matrix of normalized eigen vectors.
- $\boldsymbol{\Lambda}$ is diagonal matrix of ordered eigen values.

Finding the following axes

Second best axis

Find Δ_2 with dimension 1, director vector $\mathbf{u}_2$ orthogonal to Δ_1 solving

$$\underset{\mathbf{u}_2 \in \mathbb{R}^p}{\text{maximize}}\, I_{\Delta_2^\perp} = \mathbf{u}_2^\top \hat{\boldsymbol{\Sigma}} \mathbf{u}_2, \quad \text{with } \|\mathbf{u}_2\| = 1, \mathbf{u}_1^\top \mathbf{u}_2 = 0.$$

$\rightsquigarrow$ $\mathbf{u}_2$ is the second eigen vector of $\hat{\boldsymbol{\Sigma}}$ with eigen value λ_2

And so on!

PCA is roughly a matrix factorisation problem

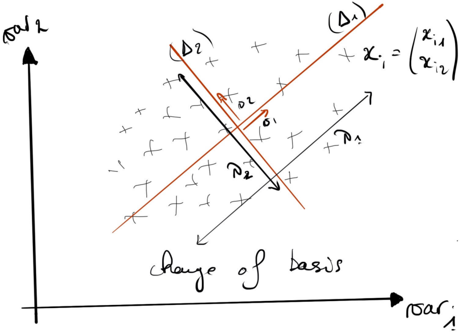
$$\hat{\boldsymbol{\Sigma}} = \mathbf{U} \boldsymbol{\Lambda} \mathbf{U}^\top, \quad \mathbf{U} = \begin{pmatrix} \mathbf{u}_1 & \mathbf{u}_2, & \dots & \mathbf{u}_p \end{pmatrix}, \quad \boldsymbol{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_p)$$

- $\mathbf{U}$ is an orthogonal matrix of normalized eigen vectors.
- $\boldsymbol{\Lambda}$ is diagonal matrix of ordered eigen values.

Interpretation in $\mathbb{R}^p$

$\mathbf{V}$ describes a new orthogonal basis and a rotation of data in this basis
$\rightsquigarrow$ PCA is an appropriate rotation on axes that maximizes the variance

$$
\left\{
\begin{array}{ccccc}
\Delta_1 & \oplus & \dots & \oplus & \Delta_p \\
\mathbf{u}_1 & \perp & \dots & \perp & \mathbf{u}_2 \\
\lambda_1 & > & \dots & > & \lambda_p \\
I_{\Delta_1^\perp} & > & \dots & > & I_{\Delta_p^\perp}
\end{array}
\right.
$$



Outline

Principal Component Analysis

Outline

Principal Component Analysis

Contribution of each axis and quality of the representation

Δ_k is carrying inertia/variance defined by its orthogonal, thus

$$I_T = I_{\Delta_1^\perp} + \cdots + I_{\Delta_p^\perp} = \lambda_1 + \cdots + \lambda_p$$

Relative contribution of axis k

$$\text{contrib}(\Delta_k) = \frac{\lambda_k}{\sum_{k=1}^p \lambda_j} = \frac{\lambda_k}{\text{trace}\left(\hat{\Sigma}\right)} \times 100$$

$\rightsquigarrow$ Percentage of explained inertia/variance explained

Global quality of the representation on the first k axes

$$\text{contrib}(\Delta_1, \ldots, \Delta_k) = \frac{\lambda_1 + \cdots + \lambda_k}{\text{trace}\left(\hat{\Sigma}\right)} \times 100$$

A few axes may explain a large proportion of the total variance.
$\rightsquigarrow$ This paves the way for dimension reduction

Contribution of each axis and quality of the representation

Δ_k is carrying inertia/variance defined by its orthogonal, thus

$$I_T = I_{\Delta_1^\perp} + \cdots + I_{\Delta_p^\perp} = \lambda_1 + \cdots + \lambda_p$$

Relative contribution of axis k

$$\text{contrib}(\Delta_k) = \frac{\lambda_k}{\sum_{k=1}^p \lambda_j} = \frac{\lambda_k}{\text{trace}\left(\hat{\boldsymbol{\Sigma}}\right)} \times 100$$

$\leadsto$ Percentage of explained inertia/variance explained

Global quality of the representation on the first k axes

$$\text{contrib}(\Delta_1, \dots, \Delta_k) = \frac{\lambda_1 + \cdots + \lambda_k}{\text{trace}\left(\hat{\boldsymbol{\Sigma}}\right)} \times 100$$

A few axes may explain a large proportion of the total variance.
$\leadsto$ This paves the way for dimension reduction

Contribution of each axis and quality of the representation

Δ_k is carrying inertia/variance defined by its orthogonal, thus

$$I_T = I_{\Delta_1^\perp} + \cdots + I_{\Delta_p^\perp} = \lambda_1 + \cdots + \lambda_p$$

Relative contribution of axis k

$$\text{contrib}(\Delta_k) = \frac{\lambda_k}{\sum_{k=1}^p \lambda_j} = \frac{\lambda_k}{\text{trace}\left(\hat{\boldsymbol{\Sigma}}\right)} \times 100$$

$\rightsquigarrow$ Percentage of explained inertia/variance explained

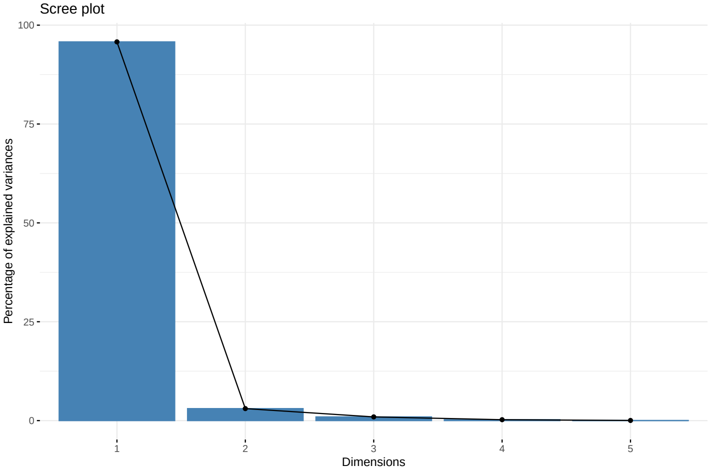
Global quality of the representation on the first k axes

$$\text{contrib}(\Delta_1, \dots, \Delta_k) = \frac{\lambda_1 + \cdots + \lambda_k}{\text{trace}\left(\hat{\boldsymbol{\Sigma}}\right)} \times 100$$

A few axes may explain a large proportion of the total variance.
$\rightsquigarrow$ This paves the way for dimension reduction

Scree plot: 'crabs'

```
crabs_pca <- select(crabs, -species, -sex) %>% FactoMineR::PCA(graph = FALSE)
fviz_eig(crabs_pca)
```



⇝ We will see during labs why everything is carried by the first axis

Outline

Principal Component Analysis

Individuals: representation in the new basis

Projection of point $\mathbf{x}_i$ axis k

The projection of $\mathbf{x}_i$ onto axis Δ_k is $c_{ik}\mathbf{u}_k$, with

$$c_{ik} = \mathbf{u}_k^\top(\mathbf{x}_i - \bar{\mathbf{x}}),$$

the coordinate of i in the basis $\mathbf{u}_k$ (along axis Δ_k).

Coordinates of i in the new basis

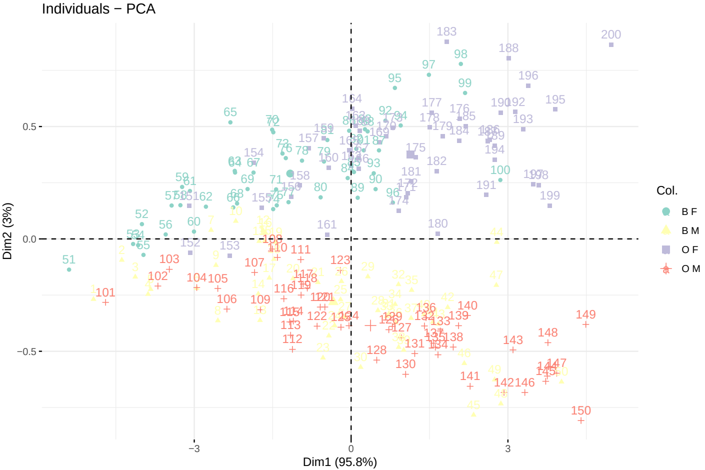
Coordinates of i in the new basis $\{\mathbf{u}1, \dots, \mathbf{u}_d\}$ is thus

$$\mathbf{c}_i = (\mathbf{U}^\top(\mathbf{x}_i - \bar{\mathbf{x}}))^\top = (\mathbf{x}_i - \bar{\mathbf{x}})^\top\mathbf{U} = \mathbf{X}_i^c\mathbf{U}, \quad \mathbf{c}_i \in \mathbb{R}^p.$$

- $\mathbf{U}$ are often the called the **loadings**, or **weights**
- $\mathbf{c}_i$ are the **scores** or **coordinates** in the new space for the individuals

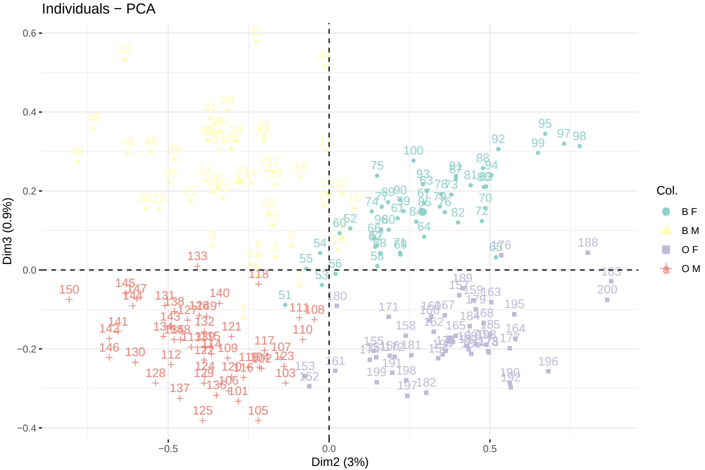
Individual visualization: projection in the new basis (1)

```
fviz_pca_ind(crabs_pca, col.ind = paste(crabs$species, crabs$sex), palette = pal)
```



Individual visualization: projection in the new basis (2)

```
fviz_pca_ind(crabs_pca, axes = c(2,3), col.ind = paste(crabs$species, crabs$sex), p
```



Warning: about distances after projection

Close projection doesn't mean close individuals!

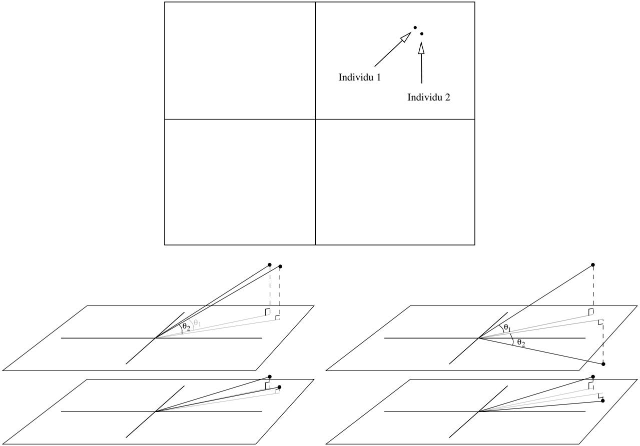


Figure: Same projections but different situations (source: E. Matzner)

⇝ Only work when individuals are well represented in the lower space

Individual: quality of the representation

Property

- An individual i is well represented by Δ_k if it is close to this axis.
- In other word, vector $\mathbf{x}_i - \bar{\mathbf{x}}$ and $\mathbf{u}_k$ are close to collinear

We use the cosine of the angle θ_{ik} between $\mathbf{x}_i - \bar{\mathbf{x}}$ and $\mathbf{u}_k$ to measure the degree of co-linearity:

$$\cos^2(\theta_{ik}) = \frac{\left(\mathbf{u}_k^\top(\mathbf{x}_i - \bar{\mathbf{x}})\right)^2}{\|\mathbf{x}_i - \bar{\mathbf{x}}\|^2 \cancel{\|\mathbf{u}_k\|^2}}$$

```r
factoextra::get_pca_ind(crabs_pca)$cos2 %>% head(3) %>% kable("latex")
```

Dim.1	Dim.2	Dim.3	Dim.4	Dim.5
0.9961694	0.0029565	0.0006132	6.29e-05	1.98e-04
0.9994582	0.0004598	0.0000800	1.60e-06	5.00e-07
0.9980940	0.0016699	0.0000663	8.50e-05	8.48e-05

Individual: contribution to an axis

Property

- Inertia "explained" by Δ_k is inertia of $\Delta_k^\perp$
- $I_{\Delta_k^\perp} = n^{-1} \sum_{i=1}^n \text{dist}^2(\Delta_k^\perp, \mathbf{x}_i)$

Contribution of $\mathbf{x}_i$ to axis Δ_k is the proportion of variance/inertia carried by individual i:

$$\text{contr}(\mathbf{x}_i) = \frac{n^{-1}\text{dist}^2(\Delta_k^\perp, \mathbf{x}_i)}{I_{\Delta_k^\perp}} = \frac{\left(\mathbf{u}_k^\top(\mathbf{x}_i - \bar{\mathbf{x}})\right)^2}{n\lambda_k}$$

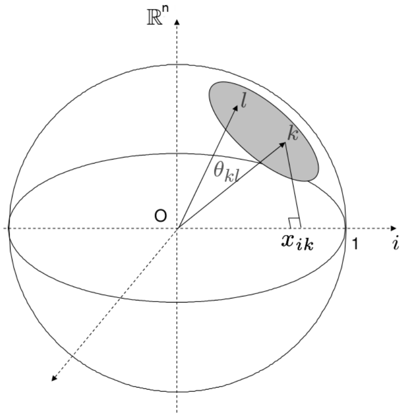
```
factoextra::get_pca_ind(crabs_pca)$contr %>% head(3) %>% kable("latex")
```

Dim.1	Dim.2	Dim.3	Dim.4	Dim.5
2.535166	0.2375409	0.1602617	0.0688010	1.4097141
2.008687	0.0291717	0.0165027	0.0013421	0.0027214
1.779751	0.0940074	0.0121362	0.0651696	0.4231593

Outline

Principal Component Analysis

Cloud of variables in $\mathbb{R}^n$



Direct equivalence between geometry and statistics (collinearity $\equiv$ correlation)

$$\cos(\theta_{kl}) = \frac{\langle \mathbf{x}^k, \mathbf{x}^\ell \rangle}{\|\mathbf{x}^k\| \|\mathbf{x}^\ell\|} = \rho(\mathbf{x}^k, \mathbf{x}^\ell)$$

Principal Components

Dual representation

A symmetric reasoning can be made in $\mathbb{R}^n$ for the variables, like with the individuals in $\mathbb{R}^p$.
$\leadsto$ New axes are linear combinason of the original variables, which can be seen as **new variables** in the new latent space

Principal component

It is the linear combinason formed by the orginal variables with weights given by the loadings $\mathbf{u}_k$

$$\mathbf{f}_k = \sum_{j=1}^{p} \mathbf{u}_k(\mathbf{x}^j - \bar{x}_j) = \mathbf{X}^c\mathbf{u}_k, \quad \mathbf{f}_k \in \mathbb{R}^n$$

Sometimes called "factors" in factor analysis, as latent (hidden) variables.

Variable representation in the new space

Connection with original variables

- essential for interpretation
- answer to the question: how reading the axis of the individual map
- use correlation to measure connection to original variable

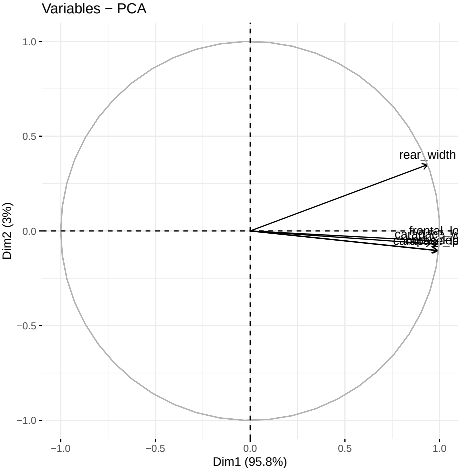
$$\mathbb{V}(\mathbf{f}_k) = \frac{1}{n}\mathbb{V}(\mathbf{X}^c\mathbf{u}_k) = \mathbf{u}_k^\top \frac{1}{n}(\mathbf{X}^c)^\top \mathbf{X}^c\mathbf{u}_k = \mathbf{u}_k^\top \hat{\mathbf{\Sigma}}\mathbf{u}_k = \lambda_k$$

$$\mathrm{cov}(\mathbf{f}_k, (\mathbf{x}^j - \bar{x}_j)) = \mathbf{u}_k \top \mathbf{X}^{c\top}\mathbf{X}^c e_j = \mathbf{u}_k \lambda_k e_j = \lambda_k \mathbf{u}_{kj}$$

$$\mathrm{cor}(\mathbf{f}_k, (\mathbf{x}^j - \bar{x}_j)) = \sqrt{\frac{\lambda_k}{\mathbb{V}(\mathbf{x}^j)}}\mathbf{u}_{kj}$$

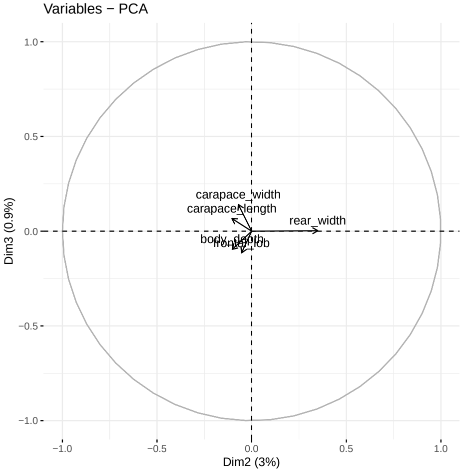
Variable vizualisation: correlation circle (1)

```r
fviz_pca_var(crabs_pca)
```



Variable vizualisation: correlation circle (2)

```
fviz_pca_var(crabs_pca, axes = c(2,3))
```



Warning: about angle after projection

Close projection doesn't mean close variable!

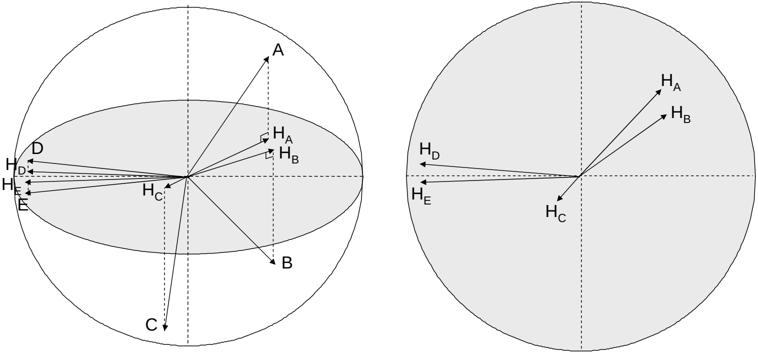


Figure: Same angle but different situations (source: J. Josse)

$\rightsquigarrow$ Only work when variables are well represented in the latent space

Variable: quality of the representation

Same story as for individuals

Property

- An variable j is well represented by Δ_k if its projection is close to $\mathbf{f}_k$.
- High collinearity means high absolute correlation and high cosine.
- use cosine to the square of the angle between the original and new variables.

$\rightsquigarrow$ The projection of j must be close to the boundardy of the correlation circle

```r
factoextra::get_pca_var(crabs_pca)$cos2 %>% head(3) %>% kable("latex")
```

	Dim.1	Dim.2	Dim.3	Dim.4	Dim.5
frontal_lob	0.9785672	0.0028712	0.0131372	0.0054085	0.0000159
rear_width	0.8775551	0.1223552	0.0000067	0.0000780	0.0000051
carapace_length	0.9835409	0.0109140	0.0044722	0.0000000	0.0010728

Variable: contribution to an axis

Similarly to individuals, we can measure the contribution of the original variables to the construction of the new ones.

```
factoextra::get_pca_var(crabs_pca)$contr %>% kable("latex")
```

	Dim.1	Dim.2	Dim.3	Dim.4	Dim.5
frontal_lob	20.43435	1.892860	28.171511	48.5702186	0.9310620
rear_width	18.32502	80.663877	0.014350	0.7006226	0.2961274
carapace_length	20.53821	7.195170	9.590266	0.0002087	62.6761450
carapace_width	20.35027	3.261487	42.584703	0.7954467	33.0080946
body_depth	20.35215	6.986605	19.639170	49.9335034	3.0885710

$\rightsquigarrow$ What do you think of the first axe ?

Outline

Principal Component Analysis

Unifying view of variables and individuals

Principal components

The full matrix of principal component connects individual coordinates to latent factors:

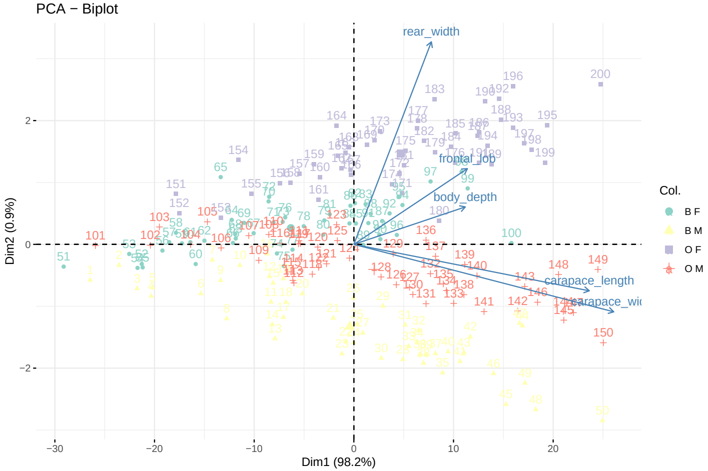
$$\mathrm{PC} = \mathbf{X}^c \mathbf{U} = \begin{pmatrix} \mathbf{f}_1 & \mathbf{f}_2 & \dots & \mathbf{f}_d \end{pmatrix} = \begin{pmatrix} \mathbf{c}_1^\top \\ \mathbf{c}_2^\top \\ \dots \\ \mathbf{c}_d^\top \end{pmatrix}$$

- new variables (latent factor) are seen column-wise
- new coordinates are seen row-wise

$\rightsquigarrow$ Everything can be interpreted on a single plot, called the biplot

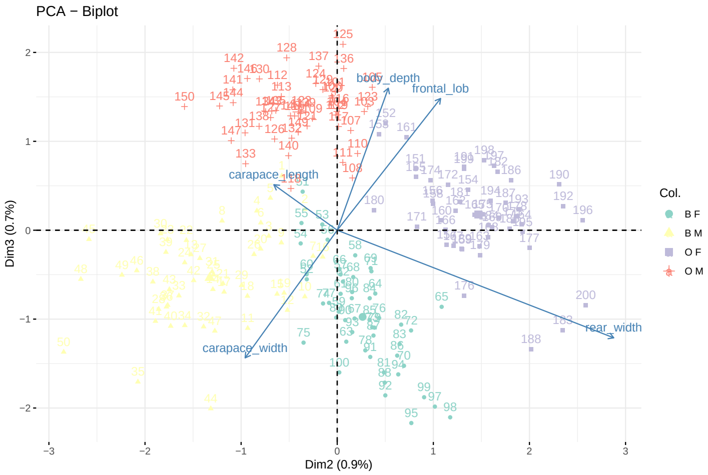
Biplot (1)

```
FactoMineR::PCA(select(crabs, -species, -sex), scale.unit = FALSE, graph = FALSE) %
  factoextra::fviz_pca_biplot(
    axes = c(1,2), col.ind = paste(crabs$species, crabs$sex), palette = pal
  )
```



Biplot (2)

```
FactoMineR::PCA(select(crabs, -species, -sex), scale.unit = FALSE, graph = FALSE) %
  factoextra::fviz_pca_biplot(
    axes = c(2,3), col.ind = paste(crabs$species, crabs$sex), palette = pal
  )
```



Reconstruction formula

Recall that $\mathbf{F} = (\mathbf{f}_1, \dots, \mathbf{f}_d)$ is the matrix of Principal components. Then,

- $\mathbf{f}_k = \mathbf{X}^c \mathbf{u}_k$ for projection on axis k
- $\mathbf{F} = \mathbf{X}^c \mathbf{U}$ for all axis.

Using orthogonality of $\mathbf{U}$, we get back the original data as follows, without loss ($\mathbf{U}^T$ performs the inverse rotation of $\mathbf{U}$):

$$\mathbf{X}^c = \mathbf{F}\mathbf{U}^\top$$

We obtain an approximation $\tilde{\mathbf{X}}^c$ (compression) of the data $\mathbf{X}^c$ by considering a subset $\mathcal{S}$ of PC, typically $\mathcal{S} = 1, \dots, K$ with $K \ll d$.

$$\tilde{\mathbf{X}}^c = \mathbf{F}_{\mathcal{S}}\mathbf{U}_{\mathcal{S}}^\top = \mathbf{X}^c \mathbf{U}_{\mathcal{S}}\mathbf{U}_{\mathcal{S}}^\top$$

$\rightsquigarrow$ This is a rank K approximation of $\mathbf{X}$ of the data the information capture by the first K axes.

Reconstruction formula

Recall that $\mathbf{F} = (\mathbf{f}_1, \dots, \mathbf{f}_d)$ is the matrix of Principal components. Then,

- $\mathbf{f}_k = \mathbf{X}^c \mathbf{u}_k$ for projection on axis k
- $\mathbf{F} = \mathbf{X}^c \mathbf{U}$ for all axis.

Using orthogonality of $\mathbf{U}$, we get back the original data as follows, without loss ($\mathbf{U}^T$ performs the inverse rotation of $\mathbf{U}$):

$$\mathbf{X}^c = \mathbf{F}\mathbf{U}^\top$$

We obtain an approximation $\tilde{\mathbf{X}}^c$ (compression) of the data $\mathbf{X}^c$ by considering a subset $\mathcal{S}$ of PC, typically $\mathcal{S} = 1, \dots, K$ with $K \ll d$.

$$\tilde{\mathbf{X}}^c = \mathbf{F}_{\mathcal{S}} \mathbf{U}_{\mathcal{S}}^\top = \mathbf{X}^c \mathbf{U}_{\mathcal{S}} \mathbf{U}_{\mathcal{S}}^\top$$

$\rightsquigarrow$ This is a rank K approximation of $\mathbf{X}$ of the data the information capture by the first K axes.

Remove size effect I

Carried by the 1st principal component

First component

$$\mathbf{f}_1 = \mathbf{X}^c \mathbf{u}_1.$$

We extract the best rank-1 approximation of $\mathbf{X}$ to remove the *size effect*, carried by the first axis, and return to the original space,

$$\tilde{\mathbf{X}}^{(1)} = \mathbf{f}_1 \mathbf{u}_1^\top.$$

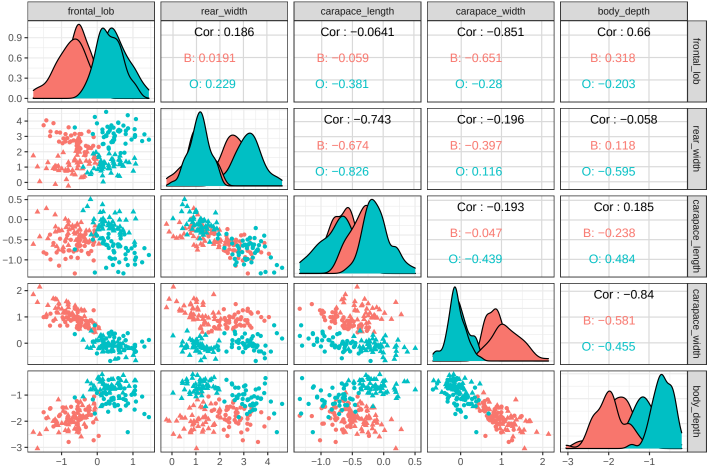
```r
attributes <- select(crabs, -sex, -species) %>% as.matrix()
u1 <- eigen(cov(attributes))$vectors[, 1, drop = FALSE]
attributes_rank1 <- attributes %*% u1 %*% t(u1)
crabs_corrected <- crabs
crabs_corrected[, 3:7] <- attributes - attributes_rank1
```

$\rightsquigarrow$ Axis 1 explains a latent effect, here the size in the case at hand, common to all attributes.

Remove size effect II

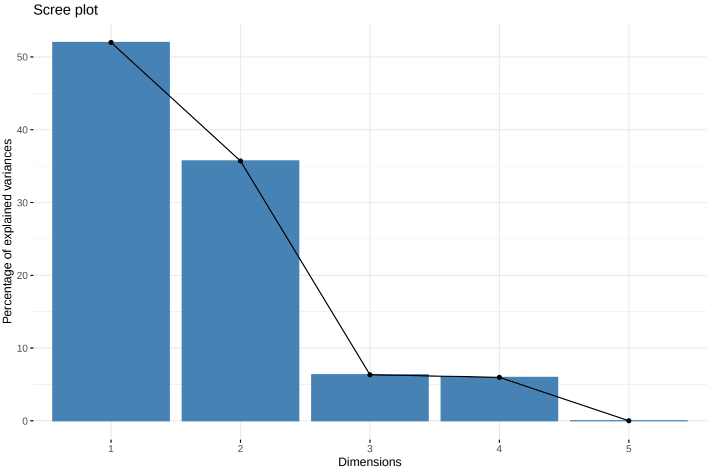
Carried by the 1st principal component

```
ggpairs(crabs_corrected, columns = 3:7, aes(colour = species, shape = sex))
```



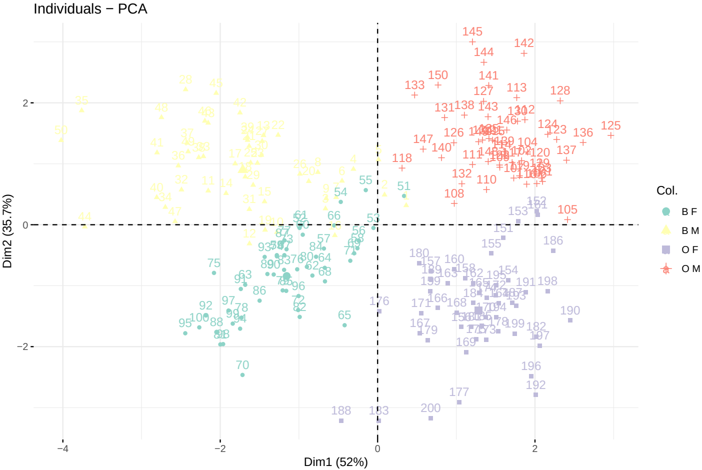
PCA on corrected data (1)

```
crabs_pca_corrected <- select(crabs_corrected, -species, -sex) %>% FactoMineR::PCA(
fviz_eig(crabs_pca_corrected)
```



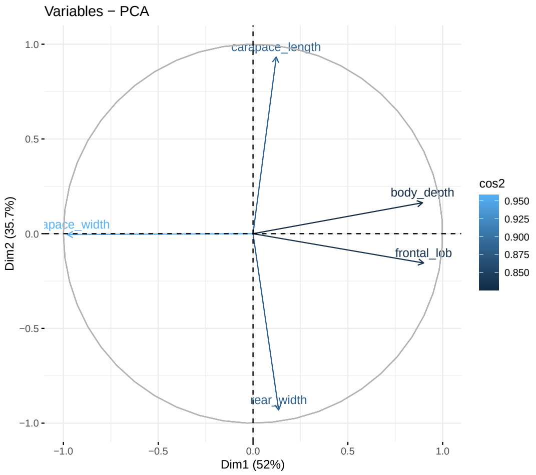
PCA on corrected data (2)

```
fviz_pca_ind(crabs_pca_corrected, col.ind = paste(crabs_corrected$species, crabs_co
```



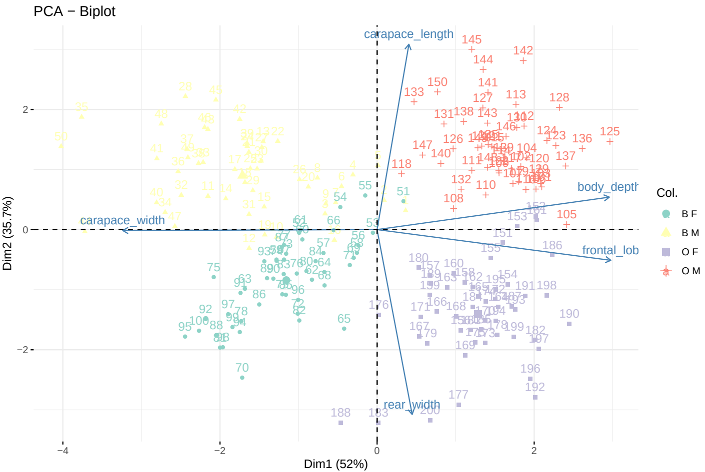
PCA on corrected data (3)

```
fviz_pca_var(crabs_pca_corrected, col.var = 'cos2')
```

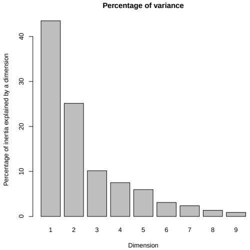


PCA on corrected data (3)

```
fviz_pca_biplot(crabs_pca_corrected, col.ind = paste(crabs_corrected$species, crabs
```



Choosing the number of components

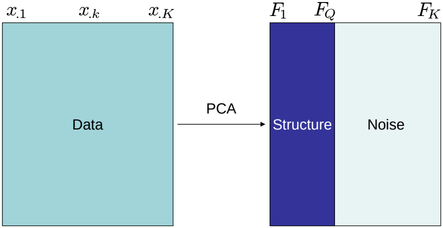


Various solutions, open question

Scree plot, test on eigenvalues, confidence interval, cross-validation, generalized cross-validation, etc.

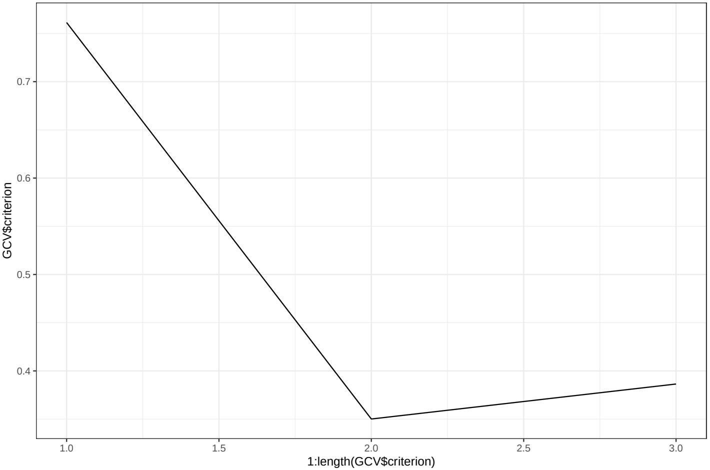
Objectives

- Interpretation
- Separate structure and noise
- Data compression



Example: Generalized Cross Validation

```
GCV <- select(crabs_corrected, -species, -sex) %>%
  FactoMineR::estim_ncp(ncp.min = 1, ncp.max = 3)
qplot(1:length(GCV$criterion), GCV$criterion, geom = "line") + labs("number of axis
```



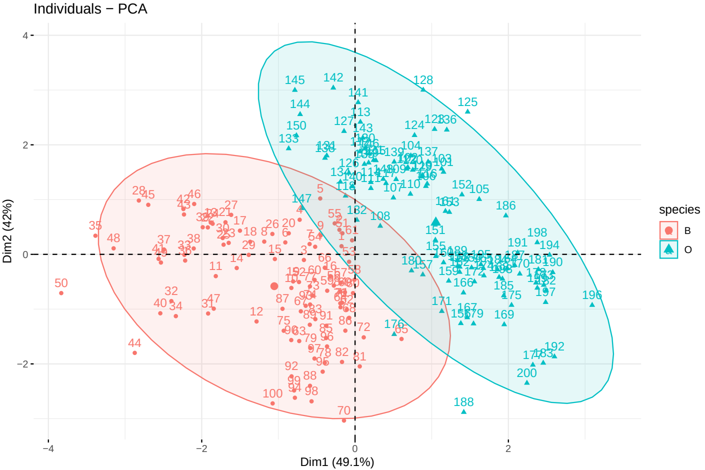
Supplementary information

- continuous variables: projection (correlation with dimensions)
- observations: projection
- categorical variables: projection of the categories at the barycentre of the observations which take the categories

```
crabs_pca_corrected <- crabs_corrected %>% FactoMineR::PCA(graph = FALSE, quanti.su
```

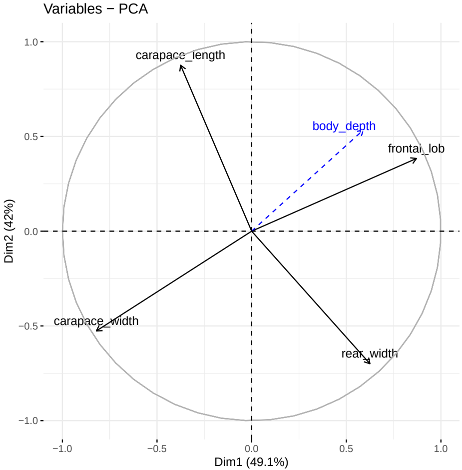
Supplementary information: example (1)

```r
factoextra::fviz_pca_ind(crabs_pca_corrected, habillage = "species", col.ind.sup =
```



Supplementary information: example (2)

```
factoextra::fviz_pca_var(crabs_pca_corrected)
```



Description of dimensions

Using continuous variables

- correlation between variable and the principal components
- sort correlation coefficients and give significant ones (rought tests)

Using categorical variables

One-way anova with the coordinates of the observations ($F_{.q}$) explained by the categorical variable

- F-test by variable
- for each category, a Student's T-test to compare the average of the category with the general mean

Description of dimensions: example

```r
FactoMineR::dimdesc(crabs_pca_corrected, axes = 1)
```

```
## $Dim.1
## $quanti
##                  correlation      p.value
## frontal_lob        0.8707523 5.928707e-63
## rear_width         0.6248516 4.683973e-23
## body_depth         0.5898360 3.935692e-20
## carapace_length   -0.3755928 4.244401e-08
## carapace_width    -0.8206976 5.086379e-50
##
## $quali
##                R2      p.value
## species 0.5653531 1.124006e-37
## sex     0.2446104 9.801298e-14
##
## $category
##             Estimate      p.value
## species=O  1.0535355 1.124006e-37
## sex=F      0.6929897 9.801298e-14
## sex=M     -0.6929897 9.801298e-14
## species=B -1.0535355 1.124006e-37
##
## attr(,"class")
```

Outline

Principal Component Analysis

Reconstruction error approach

1. Construct a map Φ from the space $\mathbb{R}^d$ into a space $\mathbb{R}^{d'}$ of smaller dimension:

$$\begin{aligned} \Phi: \quad & \mathbb{R}^d \to \mathbb{R}^{d'}, d' \ll d \\ & \mathbf{x} \mapsto \Phi(\mathbf{x}) \end{aligned}$$

2. Construct $\widetilde{\Phi}$ from $\mathbb{R}^{d'}$ to $\mathbb{R}^d$ (reconstruction formula)
3. Control an error between $\mathbf{x}$ and its reconstruction $\tilde{\Phi}(\Phi(\mathbf{x}))$, e.g

$$\sum_{i=1}^{n} \left\| \mathbf{x}_i - \tilde{\Phi}(\Phi(\mathbf{x}_i)) \right\|^2$$

Reconstruction error and PCA

PCA Model

Linear model assumption

$$\mathbf{x} \simeq \boldsymbol{\mu} + \mathbf{F}_{1:d'}\mathbf{U}_{1:d'}^{\top}$$

with $\mathbf{U}$ orthonormal and no constraint on $\mathbf{F}$

Reconstruction error

In the case of PCA, then

$$\Phi(\mathbf{x}) = (\mathbf{x} - \boldsymbol{\mu})^{\top}\mathbf{U} \quad \text{and} \quad \tilde{\Phi}(\mathbf{F}) = \boldsymbol{\mu} + \mathbf{F}\mathbf{U}^{\top}$$
$$\frac{1}{n}\sum_{i=1}^{n} \|\mathbf{x}_i - (\boldsymbol{\mu} + (\mathbf{x}_i - \boldsymbol{\mu})\mathbf{U}\mathbf{U}^{\top}\|^2$$

Explicit solution: $\boldsymbol{\mu} = \bar{x}$ the empirical mean and $\mathbf{U}$ is an orthonormal basis of the space spanned by the d' first eigenvectors of the empirical covariance matrix

Non linear extensions

Two directions

1. Non linear transformation of $\mathbf{x}$ before PCA: kernel-PCA
2. Other constrains on weigths $\mathbf{U}$ or loadings $\mathbf{F}$: ICA, NMF, ...

Kernel PCA

Linear assumption after transformation, with $\mathbf{U}$ orthonormal and no constraint on $\mathbf{F}$

$$\Psi(\mathbf{x} - \boldsymbol{\mu}) \simeq \mathbf{F}_{1:d'} \mathbf{U}_{1:d'}^{\top}$$

Non negative Matrix factorisation

Linear model assumption with $\mathbf{U}$ non-negative and $\mathbf{F}$ non-negative

$$\mathbf{x} \simeq \boldsymbol{\mu} + \mathbf{F}_{1:d'} \mathbf{U}_{1:d'}^{\top}$$

Auto-encoders Find Φ and $\tilde{\Phi}$ with a neural-network!

$\rightsquigarrow$ Fit $\mathbf{U}, \mathbf{F}$ with some optimization algorithms (much more complex!)

Outline

Principal Component Analysis

Pairwise Relation

Focus on pairwise relation $\mathcal{R}(\mathbf{x}_i, \mathbf{x}_{i'})$.

Distance Preservation

- Construct a map Φ from the space $\mathbb{R}^d$ into a space $\mathbb{R}^{d'}$ of smaller dimension:

$$\begin{aligned}\Phi : \quad & \mathbb{R}^d \to \mathbb{R}^{d'}, d' \ll d \\ & \mathbf{x} \mapsto \Phi(\mathbf{x})\end{aligned}$$

$$\text{such that} \quad \mathcal{R}(\mathbf{x}_i, \mathbf{x}_{i'}) \sim \mathcal{R}'(\mathbf{x}'_i, \mathbf{x}'_{i'})$$

Multidimensional scaling

Try to preserve inner product related to the distance (e.g. Euclidean)

t-SNE – Stochastic Neighborhood Embedding

Try to preserve relations with close neighbors with Gaussian kernel