BIOSTATISTICS COURSE #4

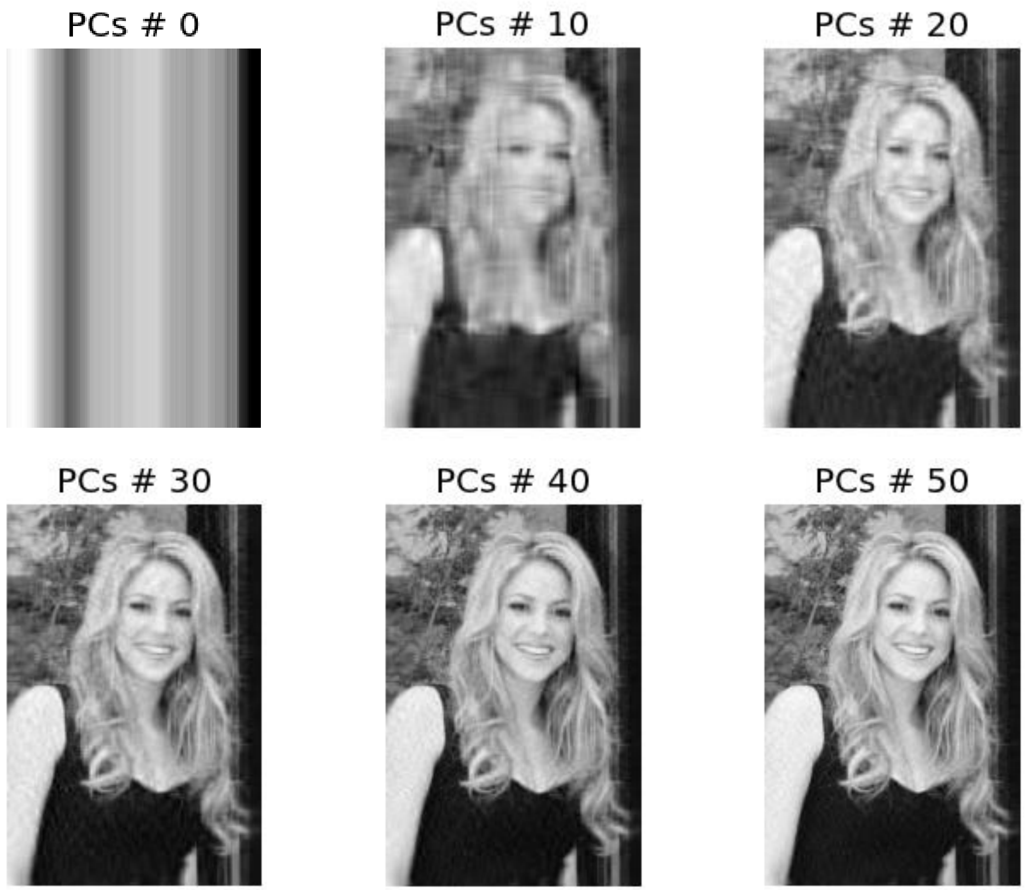
STATISTICAL MODELING #3

SEPTEMBER 2025

SUMMARY OF THE COURSE #4

INTRODUCTION 01

INTRODUCTION
DIMENSION REDUCTION



PRINCIPAL COMPONENT ANALYSIS (PCA)

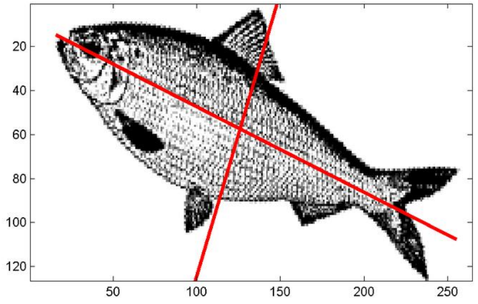
02

PRINCIPAL COMPONENT ANALYSIS (PCA)
OVERVIEW

- Principal Component Analysis emerged with **Karl PEARSON** in 1901, also called **Hotelling's transformation**

- Very powerful **reduction-dimension** statistical tool

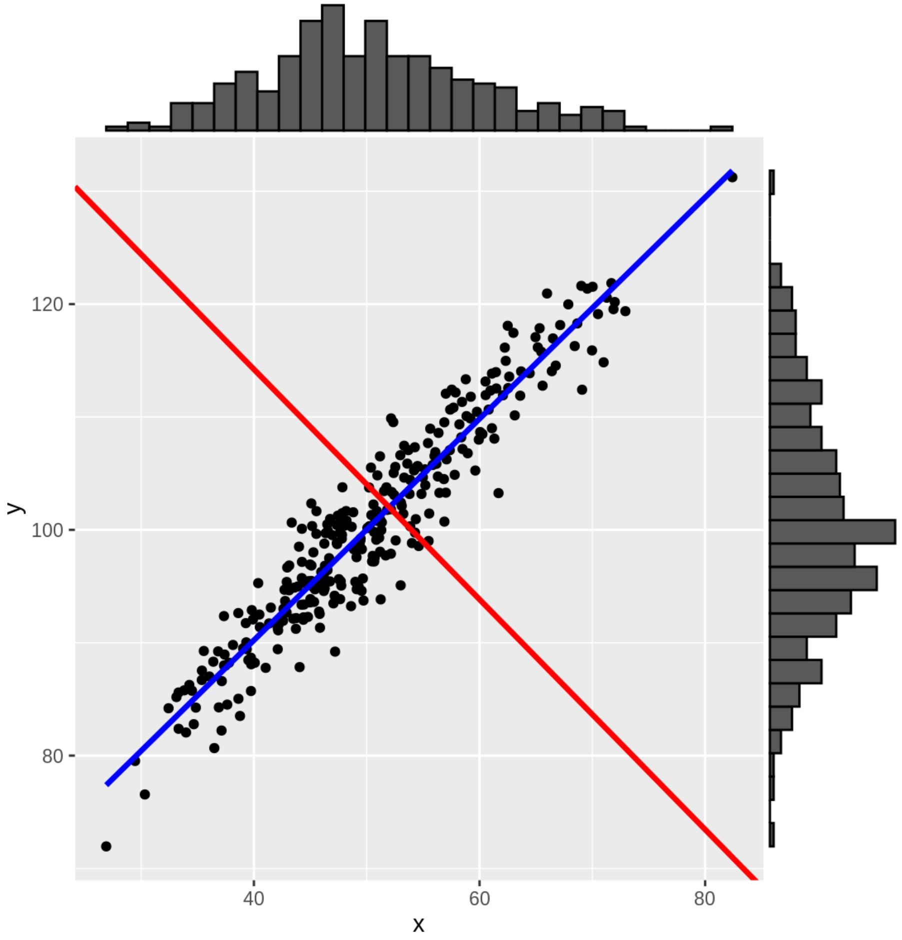
- Idea : **uncorrelate** multiple strongly linked variables in a **new multiple dimension space** (# dimensions = # variables)

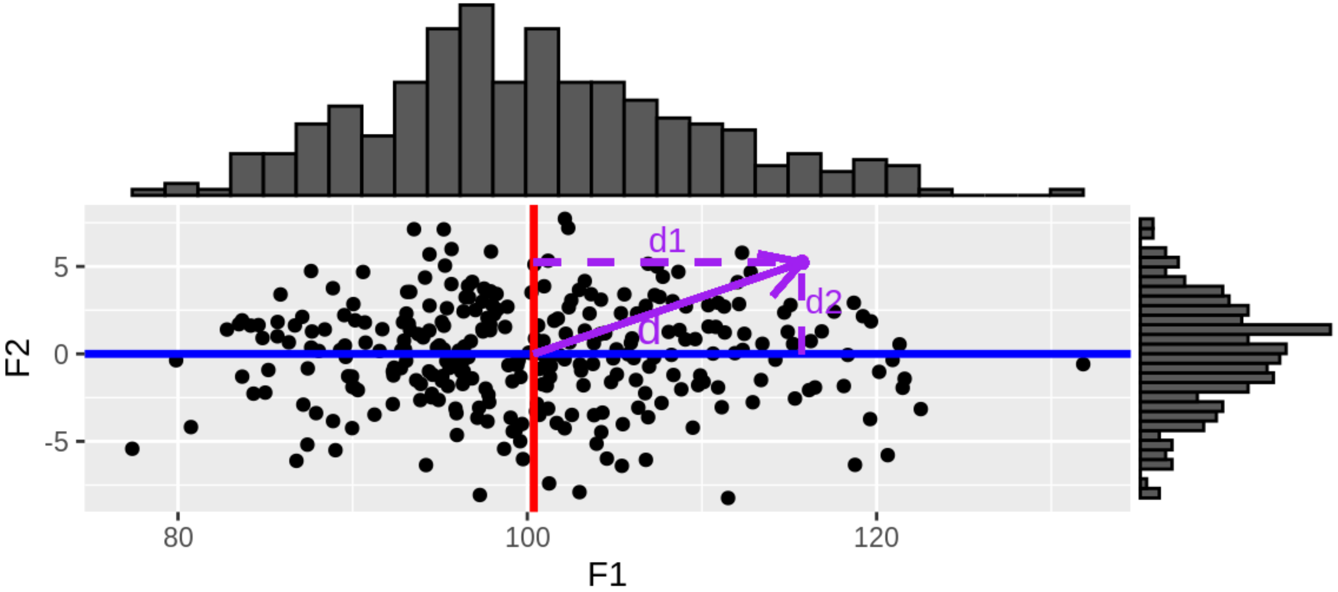


- **Many applications** : biostatistics (genetics), R&D, finance, marketing, image compression, machine-learning…

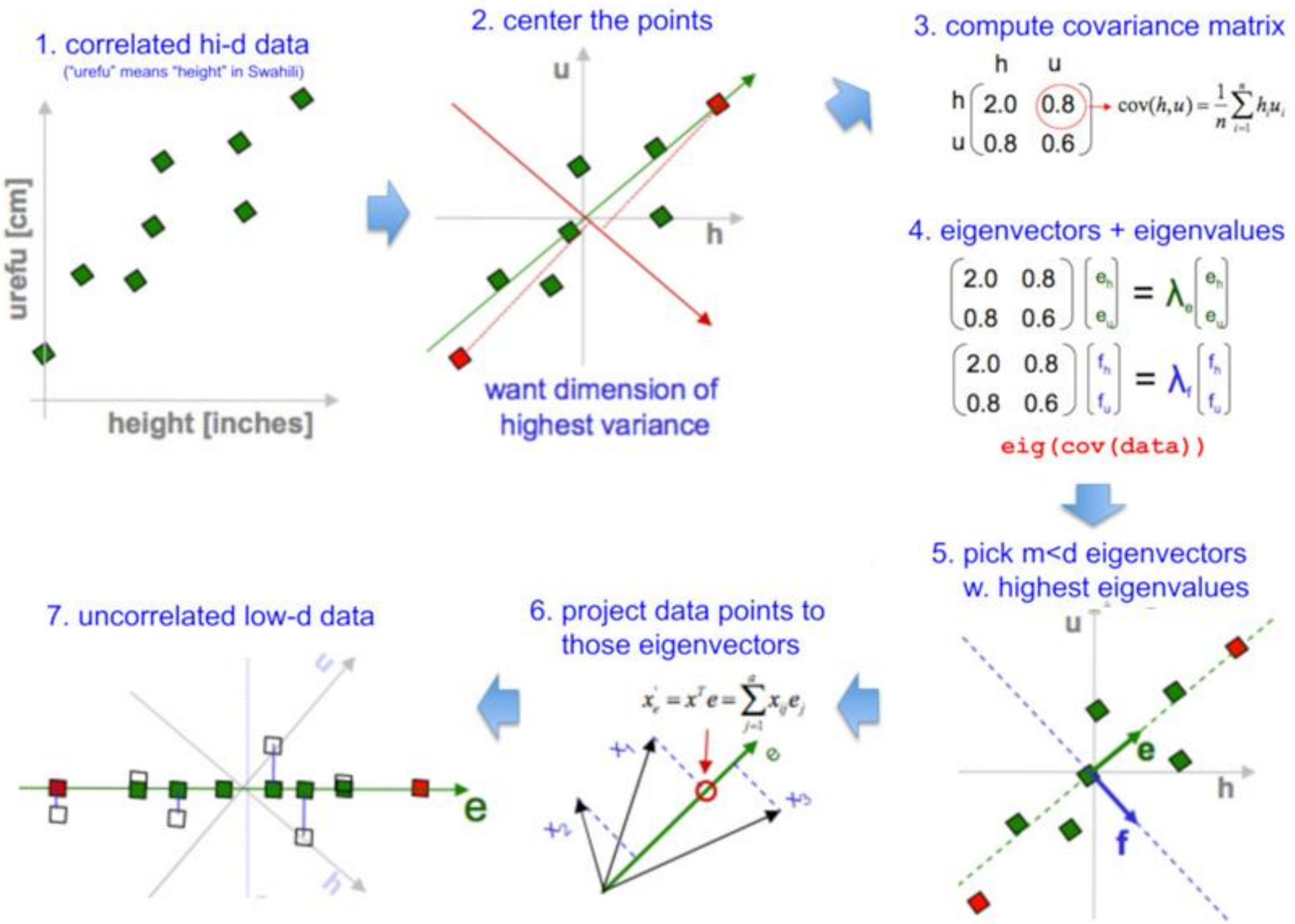
- Works with **continuous variables** (possibility to use factors for coloring points)

PRINCIPAL COMPONENT ANALYSIS (PCA)
PRINCIPAL COMPONENTS



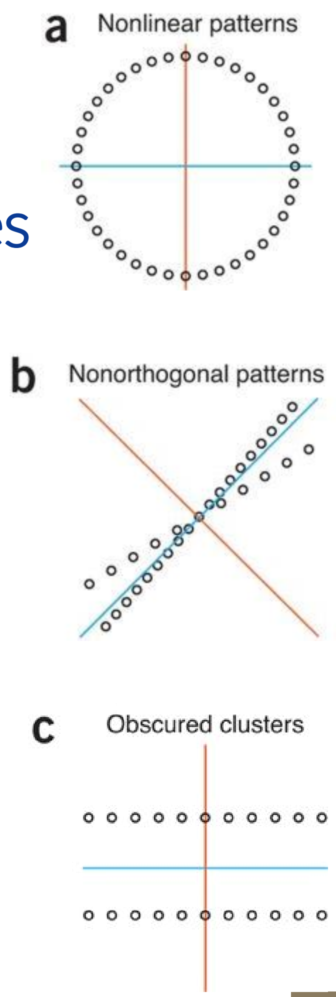


PRINCIPAL COMPONENT ANALYSIS (PCA)
WORKFLOW



PRINCIPAL COMPONENT ANALYSIS (PCA)
PRINCIPAL COMPONENTS

- Principal components (PC) are **linear combinations** of variables.
- Need to **scale values** : variables with **high variance** and/or high **values** can be too influent.
- Each PC has its own **eigenvalue** calculated on the covariance matrix
- K variables = K axes and number of spaces $= \frac{K!}{2\times(K-2)!}$
- PCA not useful when **non-linear relationship** between parameters is noticed



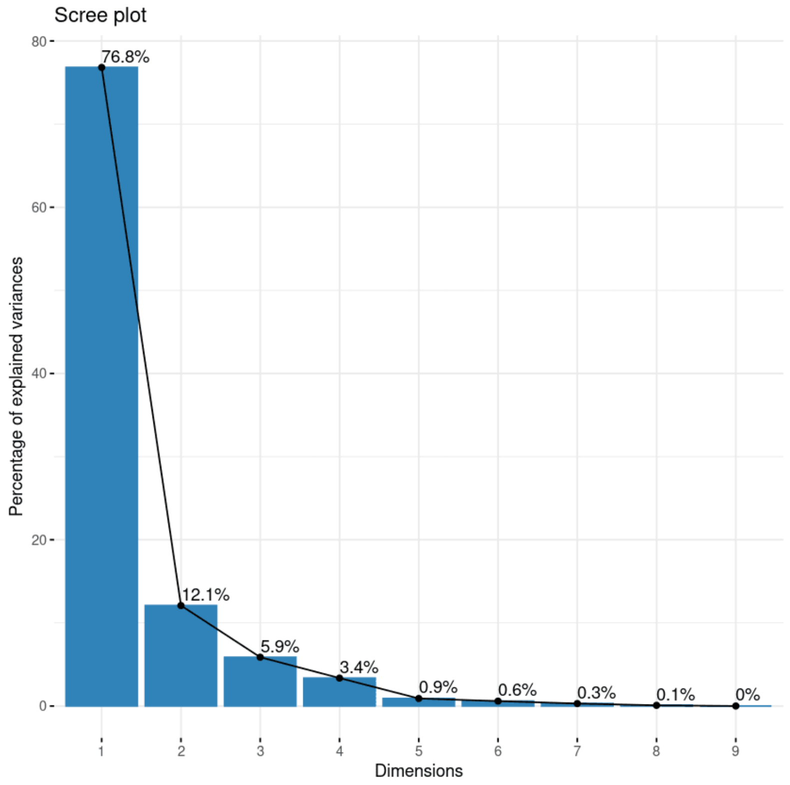
PRINCIPAL COMPONENT ANALYSIS (PCA)

PLOTS : SCREE PLOT

- Allows to visualize **eigenvalues of PCs**

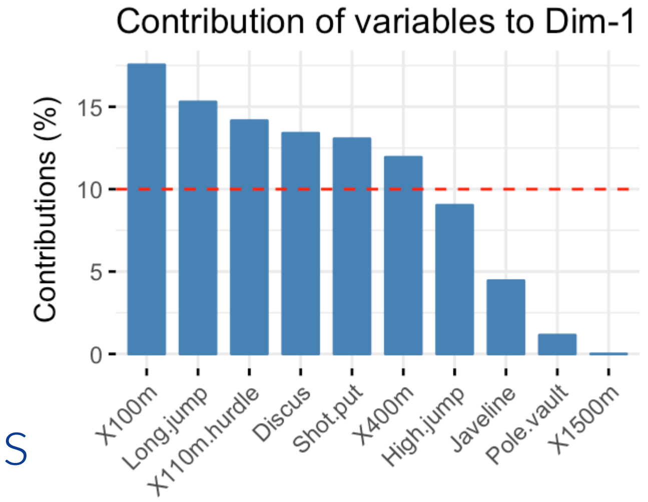
- Useful for the choice of the **number of PCs** to keep in the analysis

- In this example, the first factorial plane (PC1 + PC2) explains **almost 90% of the information contained in the dataset**

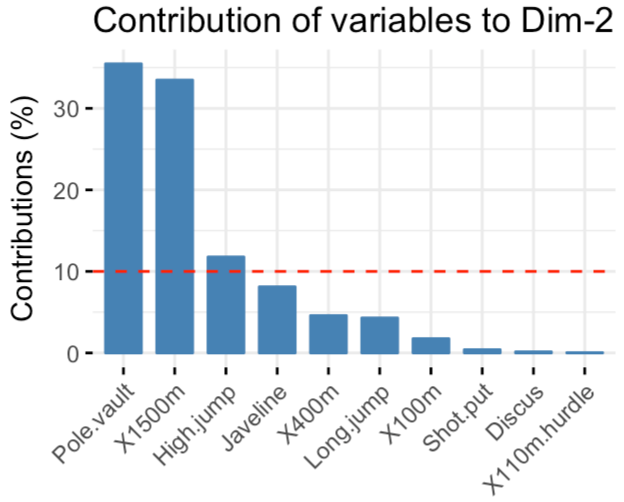


PRINCIPAL COMPONENT ANALYSIS (PCA)
PLOTS : VARIABLES CONTRIBUTION

- Allows to visualize the **most influent variables** for the building of each PC



- The **contributions are proportional** to the coordinates of each factor on the studied axis.



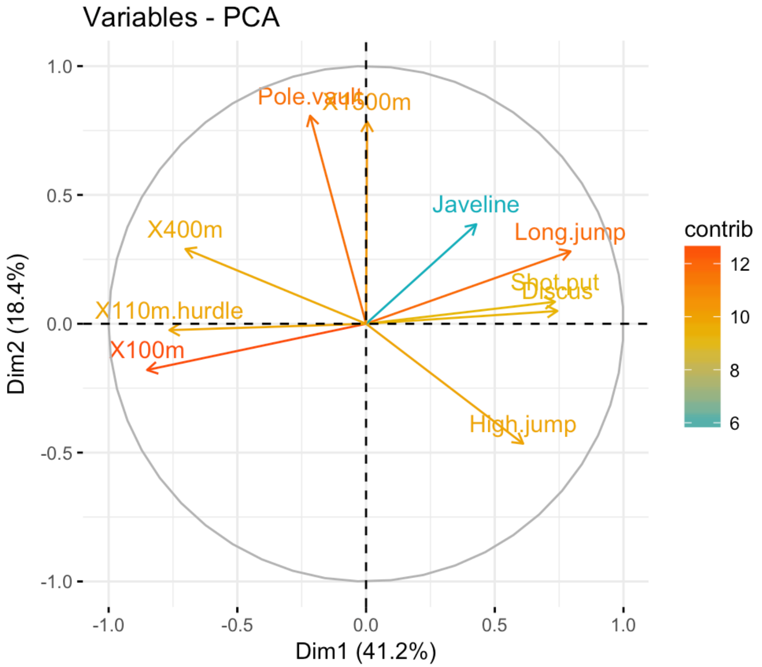
- In this example, **the second factorial plane** (PC2) **is highly impacted by the two first variables** which explains more than 60% of the variability.

PRINCIPAL COMPONENT ANALYSIS (PCA)
PLOTS : VARIABLES PLOT

- Allows to visualize the **coordinates of each variable** on the most important factorial planes.

- The coordinates are **proportional** to the contributions of each factor on each axis.

- **Opposite directions** mean significant different results for individuals.



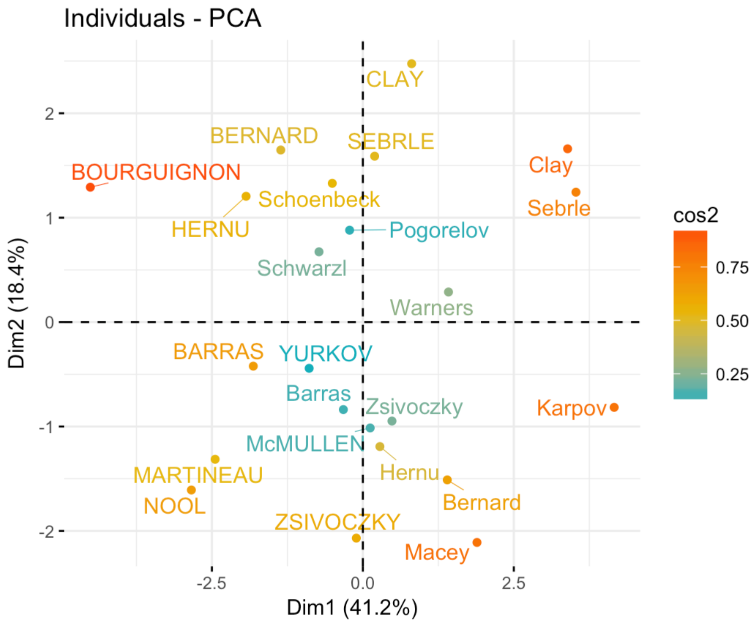
- In this example, the two extreme variables on PC1, sprint trials (100m, 100h hurdle) results are opposite to long and high jump, **meaning individuals with high performances in sprint usually don't success in jumping trials.**

PRINCIPAL COMPONENT ANALYSIS (PCA)
PLOTS : INDIVIDUALS PLOT

- Allows to visualize the **coordinates of each individual** on the most important factorial planes.

- The coordinates are **proportional** to the contributions of each factor on each axis.

- **Opposite directions** mean significant different results for individuals.



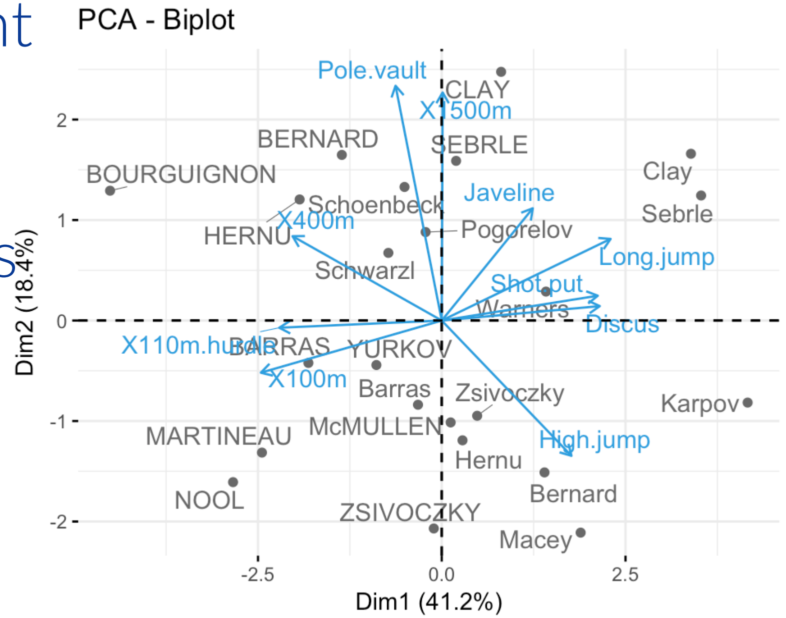
- In this example, there is no **clearly visible clusters** of individuals based on their results.

PRINCIPAL COMPONENT ANALYSIS (PCA)
PLOTS : BIPLOT

- Allows to visualize the simultaneously **coordinates of each individual and variables** on the most important factorial planes.

- The coordinates are **proportional** to the contributions of each factor on each axis.

- **Opposite directions** mean significant different results for individuals.



- In this example, there is no **clearly visible clusters** of individuals based on their results.

PRINCIPAL COMPONENT ANALYSIS (PCA)
PCA WITH R

PCA function (*FactoMiner* package)

Parameters :
- *X* = dataset with N rows and K continuous variables
- *scale.unit* = a logical value. If TRUE data is scaled $\frac{x_i - mean(x)}{sd(x)}$
- *ncp* = number of dimensions in the final result
- *quanti.sup* = indexes of the quantitative supplementary variables
- *quali.sup* = indexes of the qualitative supplementary variables
- *graph* = a logical value. If TRUE a graph is displayed

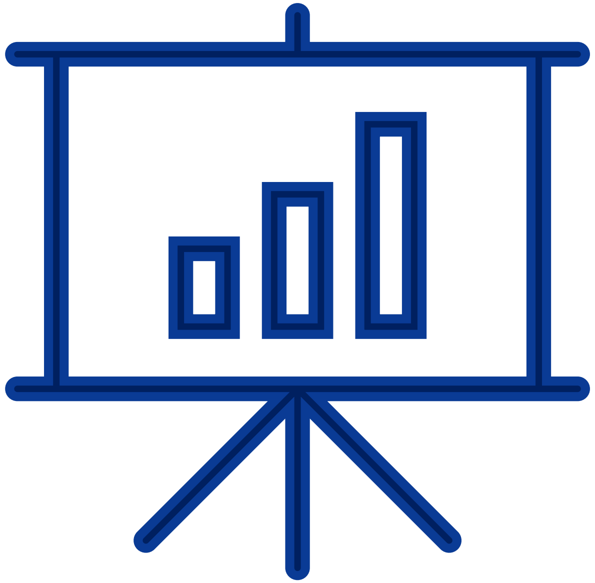
Output :
pca object containing :
- Eigenvalues of PC
- Coordinates of variables and individuals on each PC
- Contributions of variables and individuals on each PC

PRINCIPAL COMPONENT ANALYSIS (PCA)
PCA WITH R

Many additional functions in *factoextra* package (parameter : pca object : results)

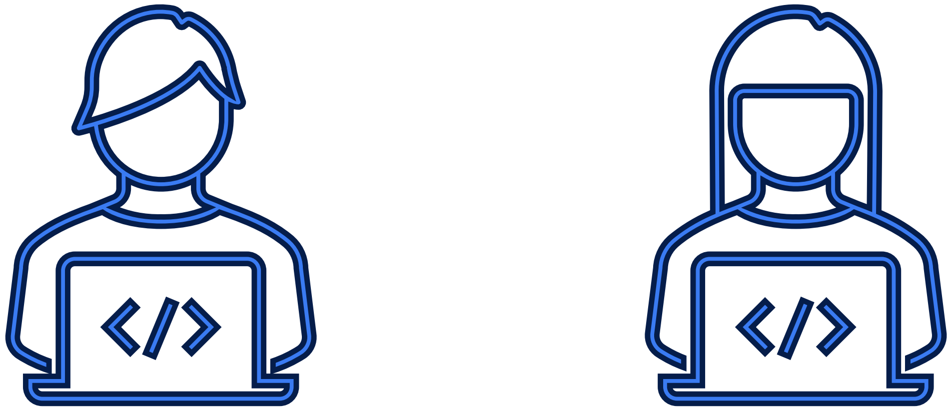
- *fviz_eig* or *fviz_screeplot* : scree plot : % of variability explained by each PC
- *fviz_contrib* : plot influence of variables or individuals in the building of PCs
- *fviz_pca_var* : plot coordinates of variables on factorial planes
- *fviz_pca_ind* : plot coordinates of individuals on factorial planes
- *fviz_pca_biplot* : plot coordinates of variables and individuals on factorial planes
- Possibility to cluster points on factorial planes (with clustering algorithms)

PRINCIPAL COMPONENT ANALYSIS (PCA)



Live demo

PRINCIPAL COMPONENT ANALYSIS (PCA)



Time to play !
(15 minutes)

OTHER FACTOR ANALYZES (FCA, MCA & FAMD)

03

FACTOR CORRESPONDENCE ANALYSIS (FCA)
OVERVIEW

- Factor Correspondence Analysis emerged with **Jean-Paul BENZECRI** in 1960.
- Goal : **study the link** (kind of correlation) between **two categorical variables**
- Works on a **contingency table or Burt table** with count data
- Instead of PCA, FCA cannot focus on variance but study **drift from independence** between the two variables.
- Same way to interpret results : **eigenvalues** are calculated with **Chi² metric** for **each modality of the variable** displayed in the columns of the contingency table

FACTOR CORRESPONDENCE ANALYSIS (FCA)
INPUT DATA

Three ways to code data :

- **Contingency tables** : count data

Contingency Table			
	Boy	Girl	Sum
like Snickers	43	30	73
doesn't like Snickers	8	19	27
Sum	51	49	100

- **Complete disjunctive table** : binary variables (0 vs 1 for each modality)

ID	Gender		Marital status		
	Male	Female	Single	Married	Divorced
1	0	1	1	0	0
2	1	0	0	1	0
3	1	0	1	0	0
4	0	1	1	0	0

- **Burt's table** : sum of counts from a complete disjunctive table (symmetric matrix)

	At Fault	Not at Fault	Female	Male	Adult	Child	Senior	Young Adult
At fault	153	0	46	107	53	64	11	25
Not at fault	0	2,112	985	1,127	1,005	411	361	746
Female	46	985	1,031	0	507	169	180	344
Male	107	1,127	0	1,234	551	306	192	491
Adult	53	1,005	507	551	1,058	0	0	0
Child	64	411	169	306	0	475	0	0
Senior	11	361	180	192	0	0	372	0
Young adult	25	746	175	185	0	0	0	360

FACTOR CORRESPONDENCE ANALYSIS (FCA)

FCA WITH R

CA function (*FactoMineR* package)

Parameters : *X* = contingency table
ncp = number of dimensions in the final result
quanti.sup = indexes of the quantitative supplementary variables
quali.sup = indexes of the qualitative supplementary variables
graph = a logical value. If TRUE a graph is displayed

Output : ca object containing :
Eigenvalues of PC
Coordinates of rows and columns on each PC
Contributions of rows and columns on each PC

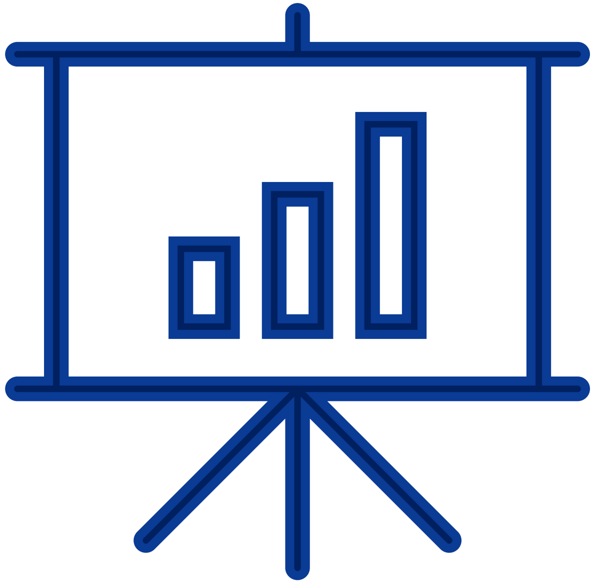
FACTOR CORRESPONDENCE ANALYSIS (FCA)

FCA WITH R

Many additional functions in *factoextra* package (parameter : ca object : results)

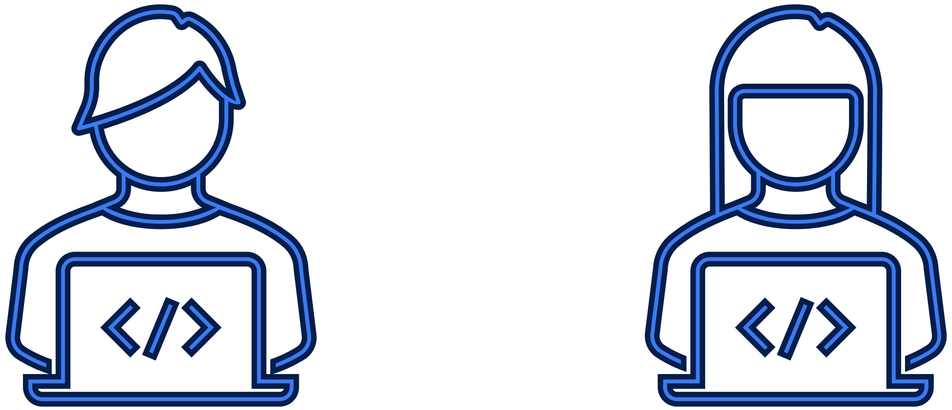
- *fviz_eig* or *fviz_screeplot* : scree plot : % of variability explained by each **PC**
- *fviz_contrib* : plot influence of **columns** or **rows modalities** in the building of PCs
- *fviz_ca_col* : plot coordinates of **columns modalities** on factorial planes
- *fviz_ca_row* : plot coordinates of **rows modalities** on factorial planes
- *fviz_ca_biplot* : plot coordinates of **columns and rows** on factorial planes
- Possibility **to cluster points on factorial planes** (with clustering algorithms)

FACTOR CORRESPONDENCE ANALYSIS (FCA)



Live demo

FACTOR CORRESPONDENCE ANALYSIS (FCA)



Time to play !
(20 minutes)

MULTIPLE CORRESPONDENCE ANALYSIS (MCA)
OVERVIEW

- Factor Correspondence Analysis emerged with **Jean-Paul BENZECRI** in 1960.
- Goal : **study the link** between several **categorical variables** (extension of FCA)
- Works on a **contingency table or Burt table** with count data
- Instead of PCA, MCA cannot focus on variance but study **drift from independence** between the variables.
- Same way to interpret results : **eigenvalues** are calculated with **Chi² metric** for **each modality of the variable** displayed in the columns of the contingency table

MULTIPLE CORRESPONDENCE ANALYSIS (MCA)
MCA WITH R

MCA function (*FactoMineR* package)

Parameters :

X = dataset with N rows and K categorical variables
ncp = number of dimensions in the final result
quanti.sup = indexes of the quantitative supplementary variables
quali.sup = indexes of the qualitative supplementary variables
graph = a logical value. If TRUE a graph is displayed

Output :

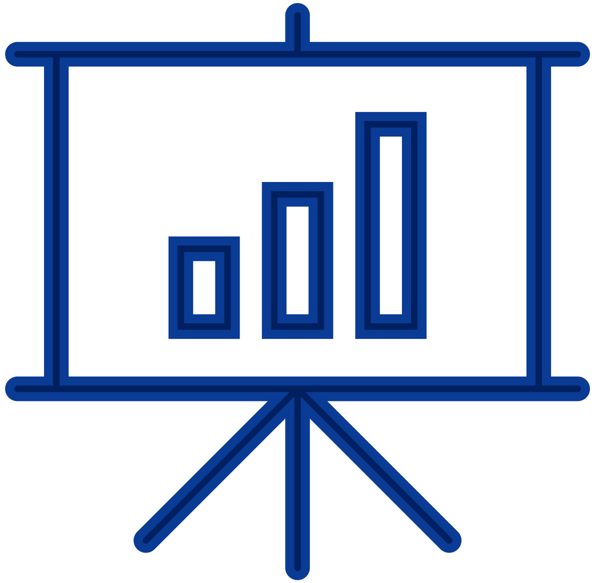
mca object containing :
Eigenvalues of PC
Coordinates of rows and columns on each PC
Contributions of rows and columns on each PC

MULTIPLE CORRESPONDENCE ANALYSIS (MCA)
MCA WITH R

Many additional functions in *factoextra* package (mca object : results)

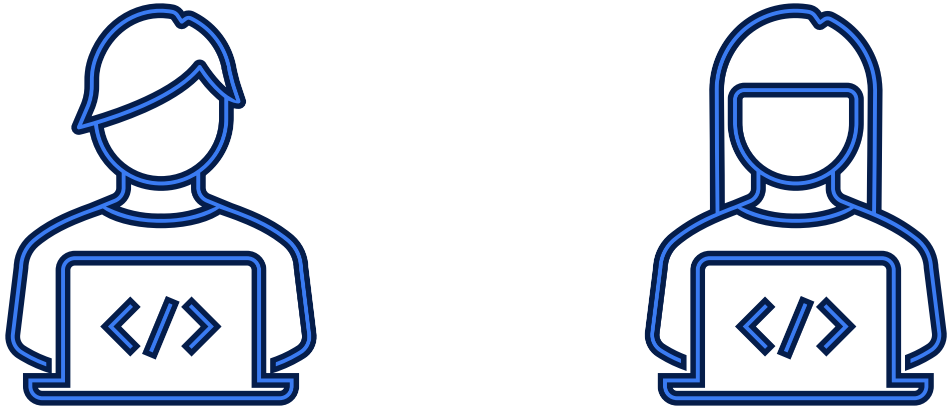
- *fviz_eig* or *fviz_screeplot* : scree plot : % of variability explained by each **PC**
- *fviz_contrib* : plot influence of **columns** or **rows modalities** in the building of PCs
- *fviz_mca_var* : plot coordinates of **variables** on factorial planes
- *fviz_mca_ind* : plot coordinates of **individuals** on factorial planes
- *fviz_mca_biplot* : plot coordinates of **columns and rows** on factorial planes
- Possibility to **cluster points** on factorial planes (with clustering algorithms)

MULTIPLE CORRESPONDENCE ANALYSIS (MCA)



Live demo

MULTIPLE CORRESPONDENCE ANALYSIS (MCA)



Time to play !
(10 minutes)

FACTOR ANALYSIS OF MIXED DATA (FAMD)
OVERVIEW

- Factor Analysis of Mixed Data emerged with **Brigitte ESCOFIER** in 1979. Work extended by **Gilbert SAPORTA** in 1990.

- Goal : **study the link** between **categorical variables** and **continuous variables**

- Works on a **raw dataset**

- A kind of **link matrix** between parameters is calculated :
  - $correlation\ coefficient\ r = \frac{\sum(x_i - \bar{x}) \times (y_i - \bar{y})}{\sqrt{(x_i - \bar{x})^2 \times (y_i - \bar{y})^2}}$ for quantitative parameters
  - $\phi^2 = \frac{\chi^2}{N}$ for categorical parameters
  - $\sqrt{r}$ for mixed variables

FACTOR ANALYSIS OF MIXED DATA (FAMD)
FAMD WITH R

FAMD function (*FactoMineR* package)

Parameters : *base* = dataset with N rows and K variables
ncp = number of dimensions in the final result
graph = a logical value. If TRUE a graph is displayed

Output : famd object containing :
Eigenvalues of PC
Coordinates of rows and columns on each PC
Contributions of rows and columns on each PC

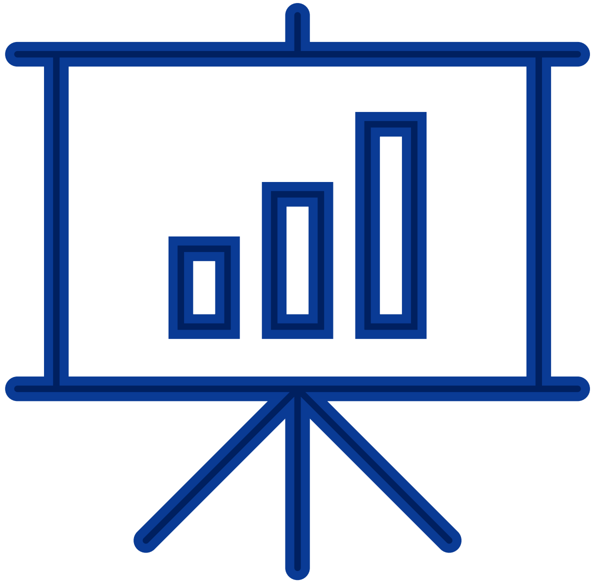
FACTOR ANALYSIS OF MIXED DATA (FAMD)

FAMD WITH R

Many additional functions (common parameter : famd object : results)

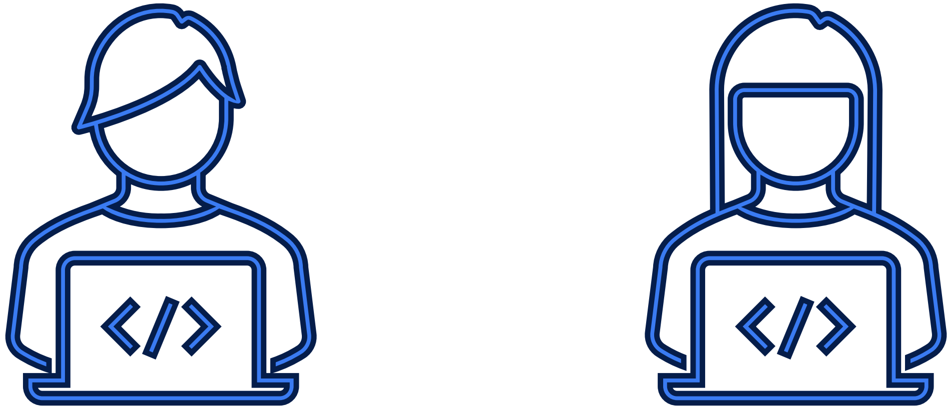
- *fviz_eig* or *fviz_screeplot* : scree plot : % of variability explained by each PC
- *fviz_contrib* : plot influence of columns or rows modalities in the building of PCs
- *fviz_famd_var* : plot coordinates of variables on factorial planes
- *fviz_famd_ind* : plot coordinates of individuals on factorial planes
- *fviz_famd_biplot* : plot coordinates of columns and rows on factorial planes
- Possibility to cluster points on factorial planes (with clustering algorithms)

FACTOR ANALYSIS OF MIXED DATA (FAMD)



Live demo

FACTOR ANALYSIS OF MIXED DATA (FAMD)

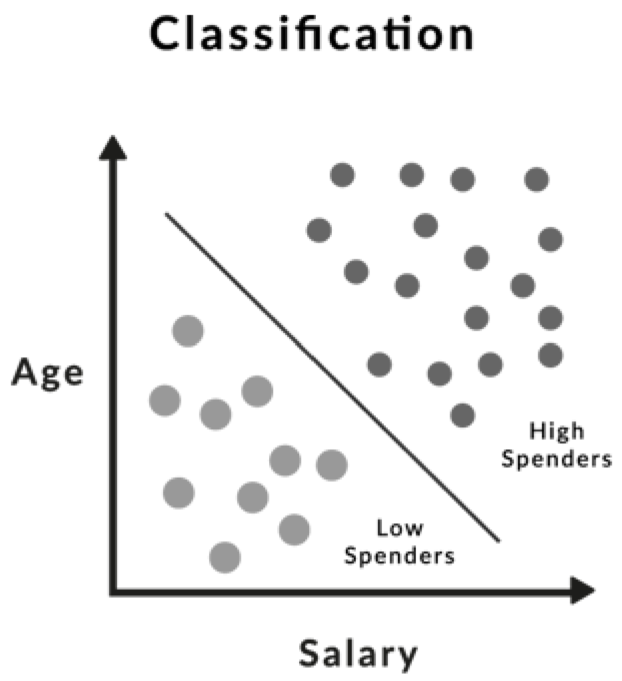


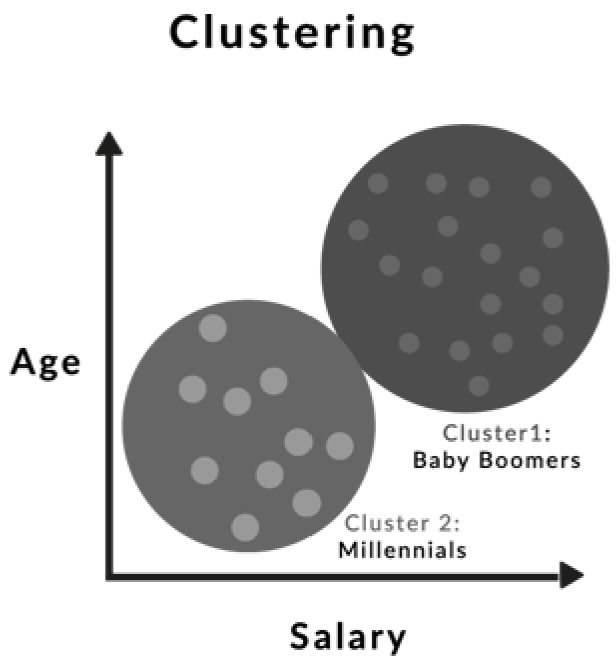
Time to play !
(20 minutes)

CLASSIFICATION & CLUSTERING

04

CLASSIFICATION & CLUSTERING
CLASSIFICATION VS CLUSTERING

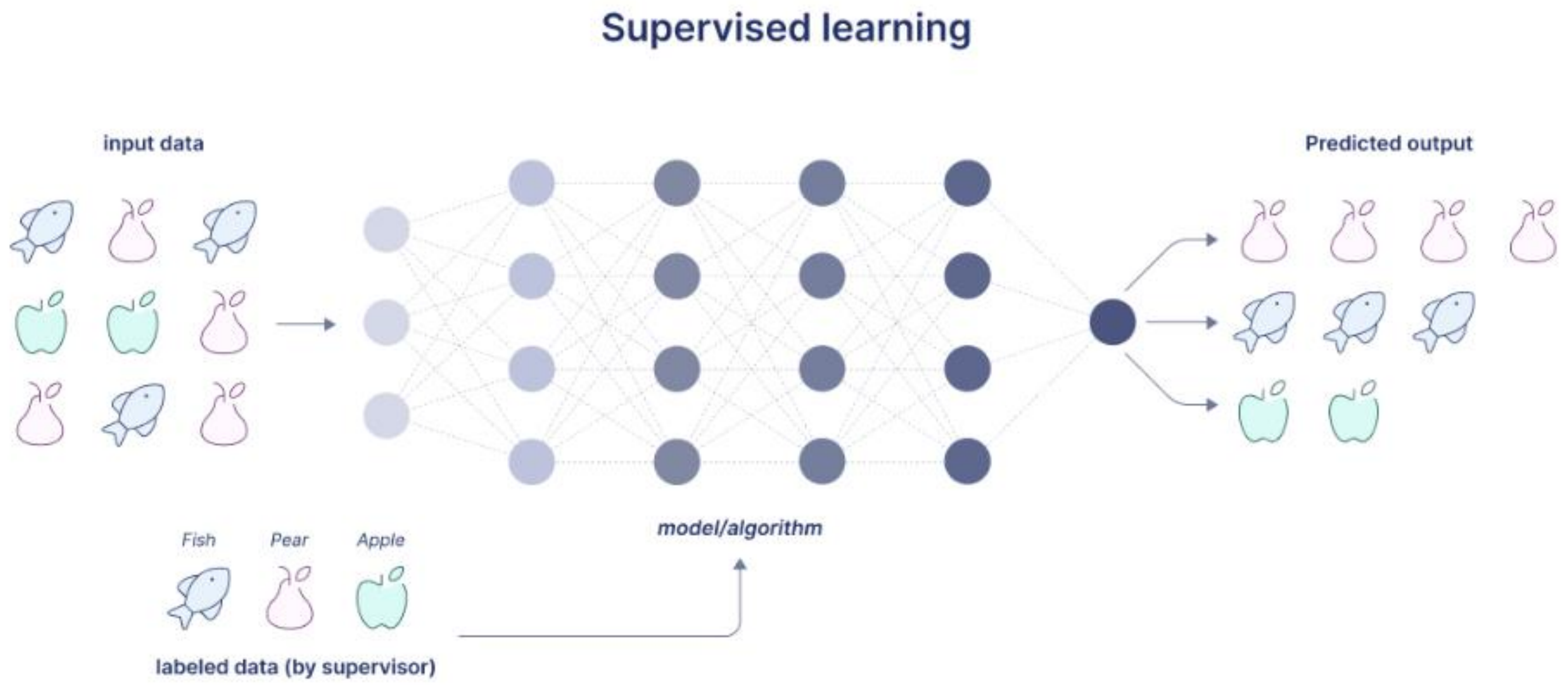




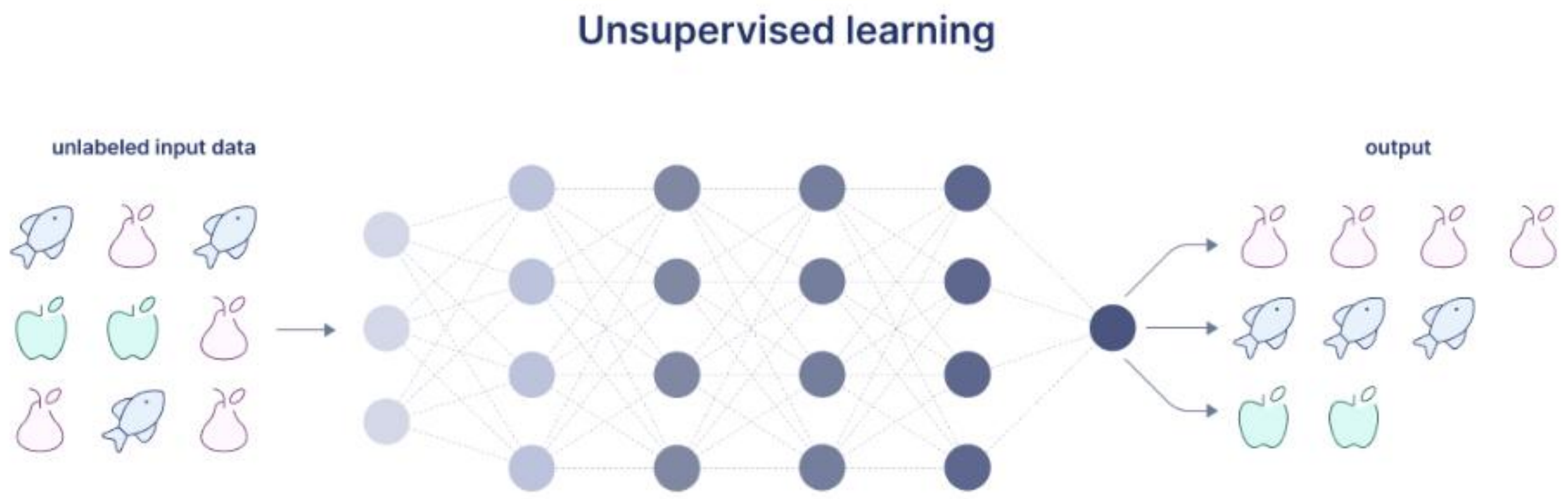
CLASSIFICATION & CLUSTERING
CLASSIFICATION VS CLUSTERING

Classification	Clustering
Uses labelled data as the input	Uses unlabelled data as the input
The output is known	The output is unknown
Uses supervised machine learning	Uses unsupervised machine learning
A training data set is provided and used to produce classifications	A training data set is not provided and used to produce clusters
Examples of algorithms: Decision-trees, Bayesian Classifiers and Support Vector Machines (SVM)	Examples of algorithms: Partition-based clustering (k-means), Hierarchical clustering (agglomerative & divisive) and DBSCAN
Can be more compex than clustering	Can be less compex than classification
Does not specify areas for improvement	Specifies areas for improvement
Two-phase	Single-phase
Boundary conditions must be specified	Boundary conditions do not always need to be specified

CLASSIFICATION & CLUSTERING
SUPERVISED VS UNSUPERVISED LEARNING



CLASSIFICATION & CLUSTERING
SUPERVISED VS UNSUPERVISED LEARNING



CLASSIFICATION & CLUSTERING

CLASSIFICATION QUALITY : CONFUSION MATRIX

Predicted / Actual	Sick	Healthy
Positive test	TRUE positive (TP)	FALSE positive (FP)
Negative test	FALSE negative (FN)	TRUE negative (TN)

$$\textbf{Specificity} = \frac{TP}{TP+FN}$$

(True Positive Rate : probability the test is positive in the sick pop)

$$\textbf{Sensitivity} = \frac{TN}{TN+FP}$$

(False Positive Rate : probability the test is negative in the healthy pop)

$$\textbf{Positive Predicted Value} = \frac{TP}{TP+FP}$$

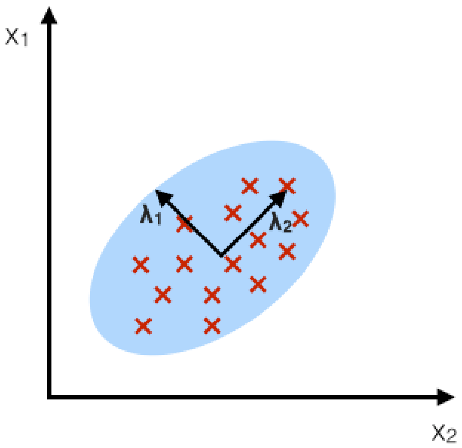
(probability the illness is present when test is positive)

$$\textbf{Negative } \boldsymbol{Predicted\ Value} = \frac{TN}{TN+FN}$$

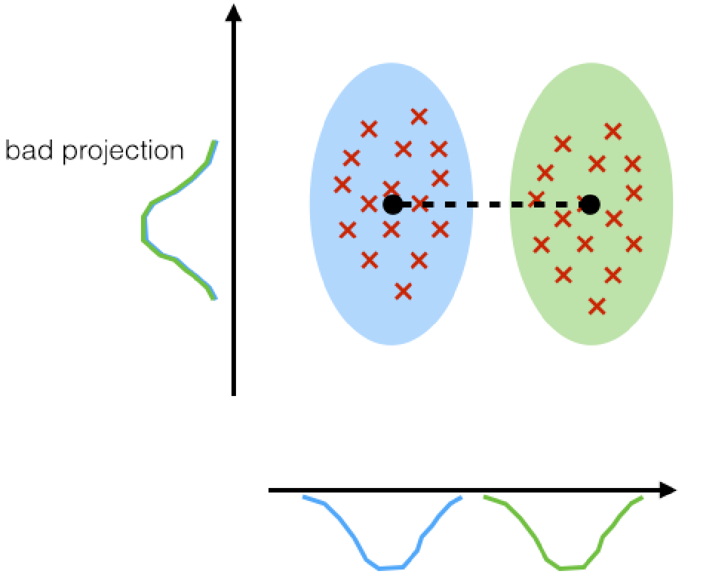
(probability the illness is absent when test is negative)

LINEAR DISCRIMINANT ANALYSIS (LDA)
INTRODUCTION

PCA:
component axes that maximize the variance



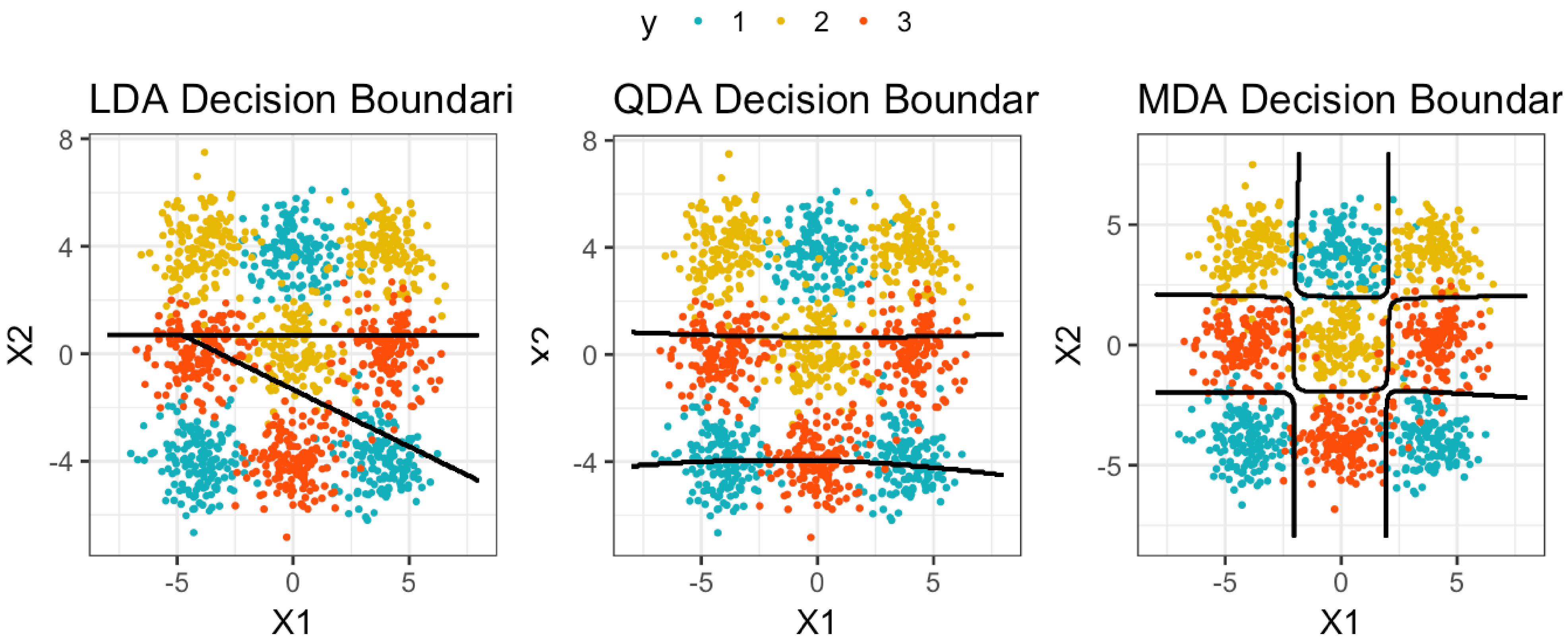
LDA:
maximizing the component axes for class-separation



LINEAR DISCRIMINANT ANALYSIS (LDA)
OVERVIEW

- Linear Discriminant Analysis emerged with **Ronald FISHER** in 1936
- Goal : find the **best linear combination** of factors which draws the most discriminant boundaries between groups.
- Works on **continuous variables** (+ 1 categorical variable) : **supervised learning**
- Assumptions : predictors are **normally distributed** in the modalities of the group variable and **homoscedasticity of variances** in groups.
- **Alternative models** : Quadratic Discriminant Analysis (QDA), Flexible Discriminant Analysis (FDA), Mixture Discriminant Analysis (MDA)

LINEAR DISCRIMINANT ANALYSIS (LDA)
LDA VS QDA VS MDA



LINEAR DISCRIMINANT ANALYSIS (LDA)

LDA WITH R

LDA function (*MASS* package)

Parameters : *data* = dataset with N rows and K variables
formula = classes ~ predictors
subset = subset of data to use
na.action = a logical value. Action to do with missing values

Output : Prior **probabilities** of groups
Group means
Coefficients of linear discriminants
Proportion of trace of each discriminant function

Machine-Learning algorithm : need to **separate dataset into train & test subsets** !

LINEAR DISCRIMINANT ANALYSIS (LDA)

LDA WITH R

ldahist function (*MASS* package)

Parameters : *data* = coordinates of observations from a LDA object
g = class to predict

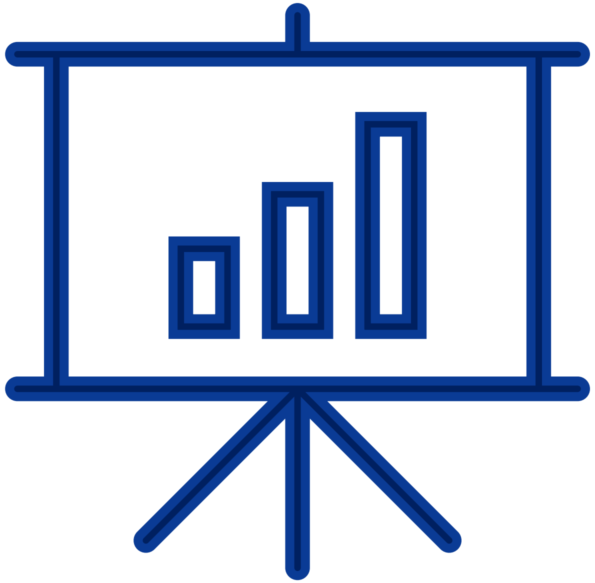
Output : A plot composed by histograms representing the classes

partimat function (*klaR* package)

Parameters : *formula* = class ~ .
data = data (train or test dataset)
method = "lda"

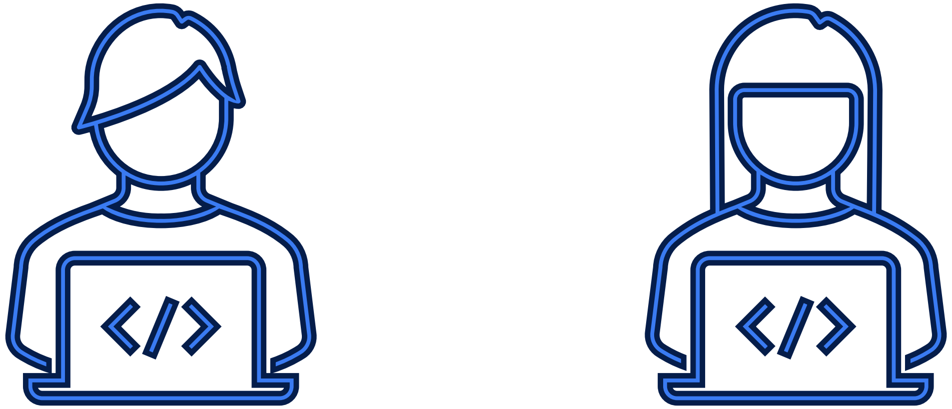
Output : Plots which represent the classification of observations in 2D figures

LINEAR DISCRIMINANT ANALYSIS (LDA)



Live demo

LINEAR DISCRIMINANT ANALYSIS (LDA)



Time to play !
(15 minutes)

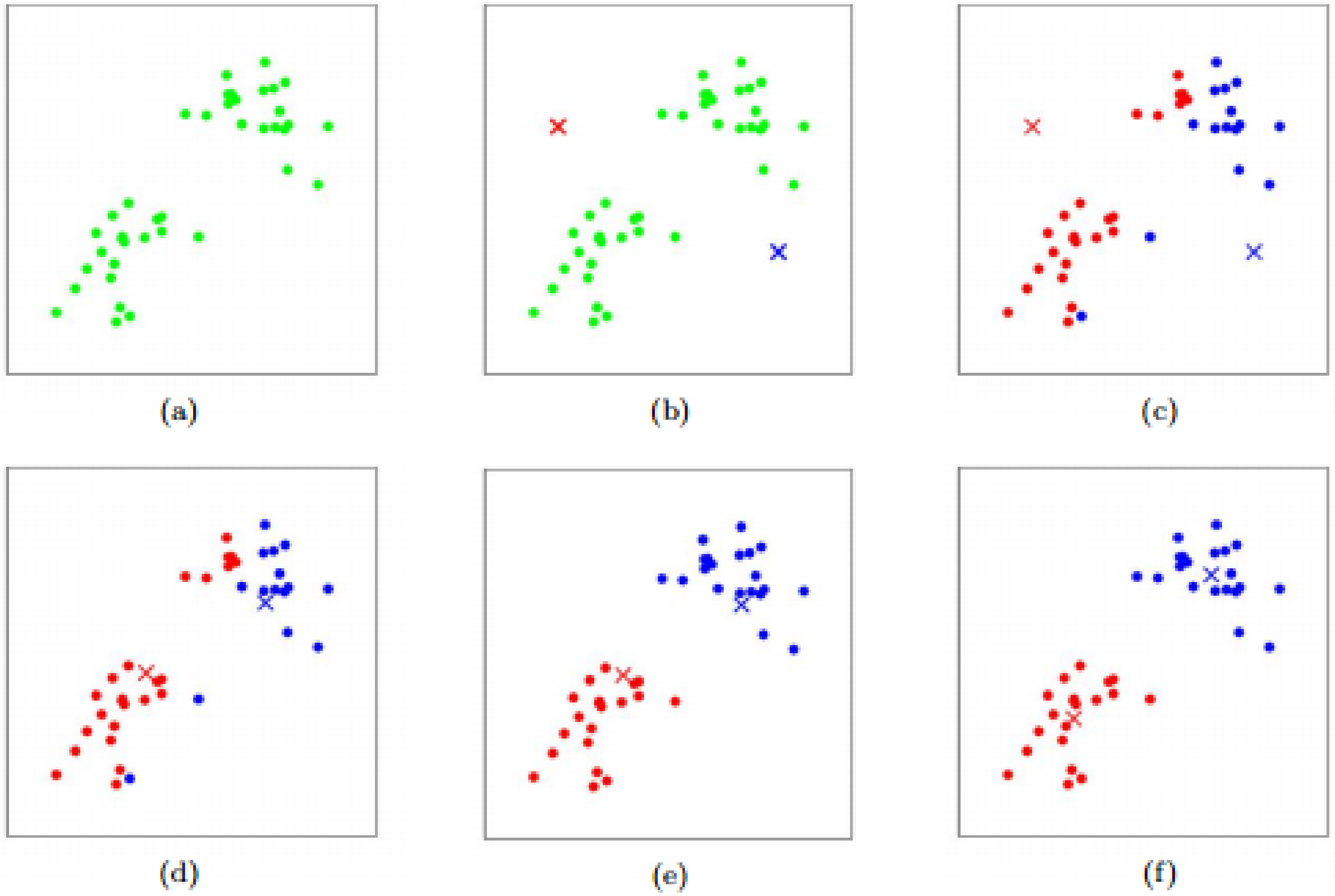
K-MEANS CLUSTERING
OVERVIEW

- K-Means algorithm invented by **Hugo Steinhaus** in 1957 from signal processing
- Goal : cluster groups of individuals into **K-groups**
- **Unsupervised learning** : the number of groups (K) is provided by user
- Based on centroids : $W(C_k) = \sum_{x_i \in C_k} (x_i - \mu_k)^2$

  x_i : individual belonging to cluster $\boldsymbol{C_k}$

  μ_k : mean value of individuals assigned to $\boldsymbol{C_k}$
- Works on **continuous variables**

K-MEANS CLUSTERING
FUNCTIONING



K-MEANS CLUSTERING
K-MEANS VS K-NEAREST NEIGHBORS (K-NN)

Feature	k-Means Clustering	k-Nearest Neighbor (k-NN)
Type of Algorithm	Unsupervised learning	Supervised learning
Purpose	Grouping similar data points into clusters	Classifying or predicting based on nearest neighbors
Data Requirements	No labeled data required	Requires labeled training data
Computational Complexity	Iterative process	Computationally intensive at prediction time
Output	Centroids and cluster assignments	Predicted labels or values

K-MEANS CLUSTERING
K-MEANS WITH R

kmeans function (*stats* package)

Parameters :

data = dataset with continuous variables
centers = K : number of clusters to build
iter.max = maximum number of iterations allowed.
nstart = number of random starting partitions (> 1)

Output :

Cluster **means**
Cluster number of each individual
Total Sum of Squares (TSS)
Between and Within Clusters Sum of Squares (BSS and WSS)

K-MEANS CLUSTERING

K-MEANS WITH R

fviz_nbclust function (*factoextra* package) allow to determine the optimal number of clusters to be generated

Parameters :

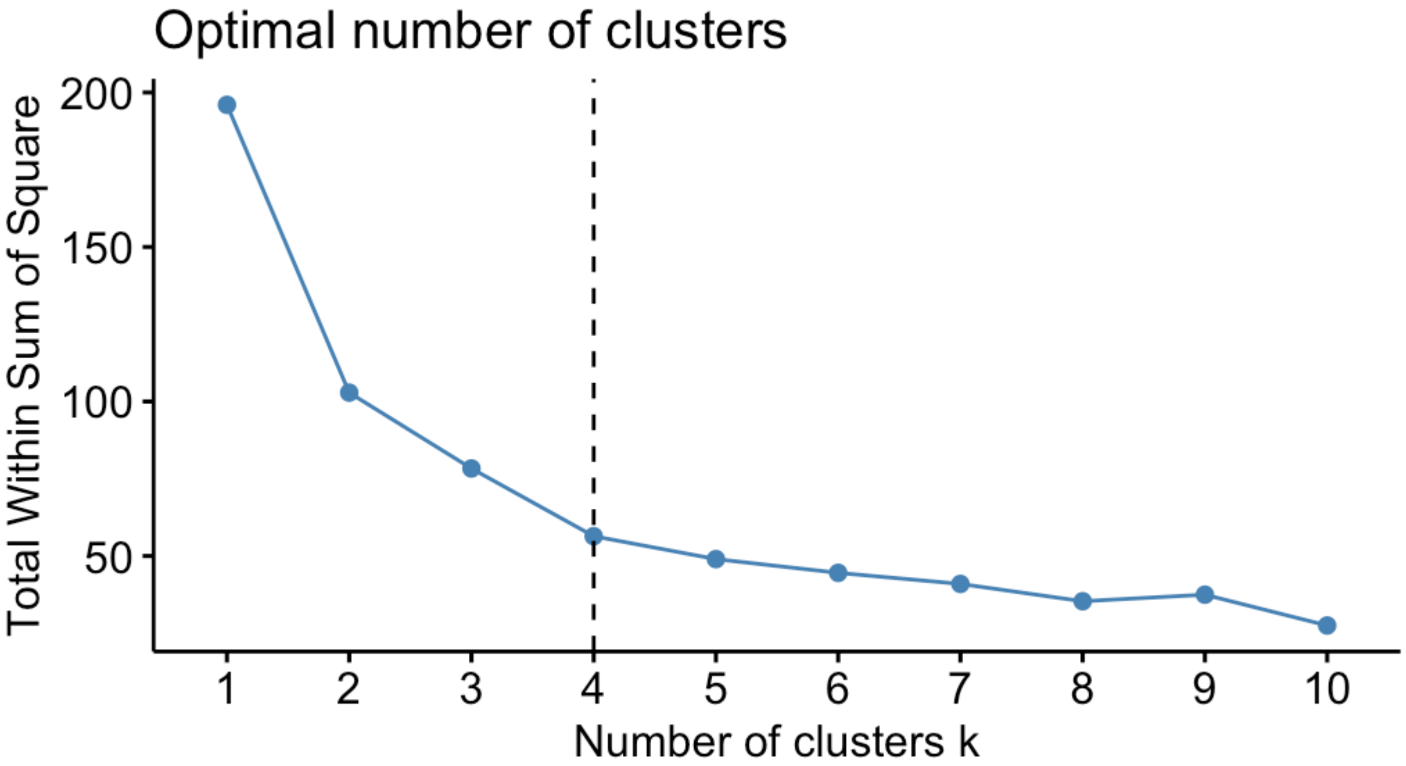
x = dataset with continuous variables
FUNcluster = kmeans
method = « *wss* »
k.max = maximum number of clusters to test
nboot = number of boostrap iterations
nstart = number of random starting partitions (> 1)

K-MEANS CLUSTERING

K-MEANS WITH R

fviz_nbclust function (*factoextra* package) allow to determine the optimal number of clusters to be generated

Output :

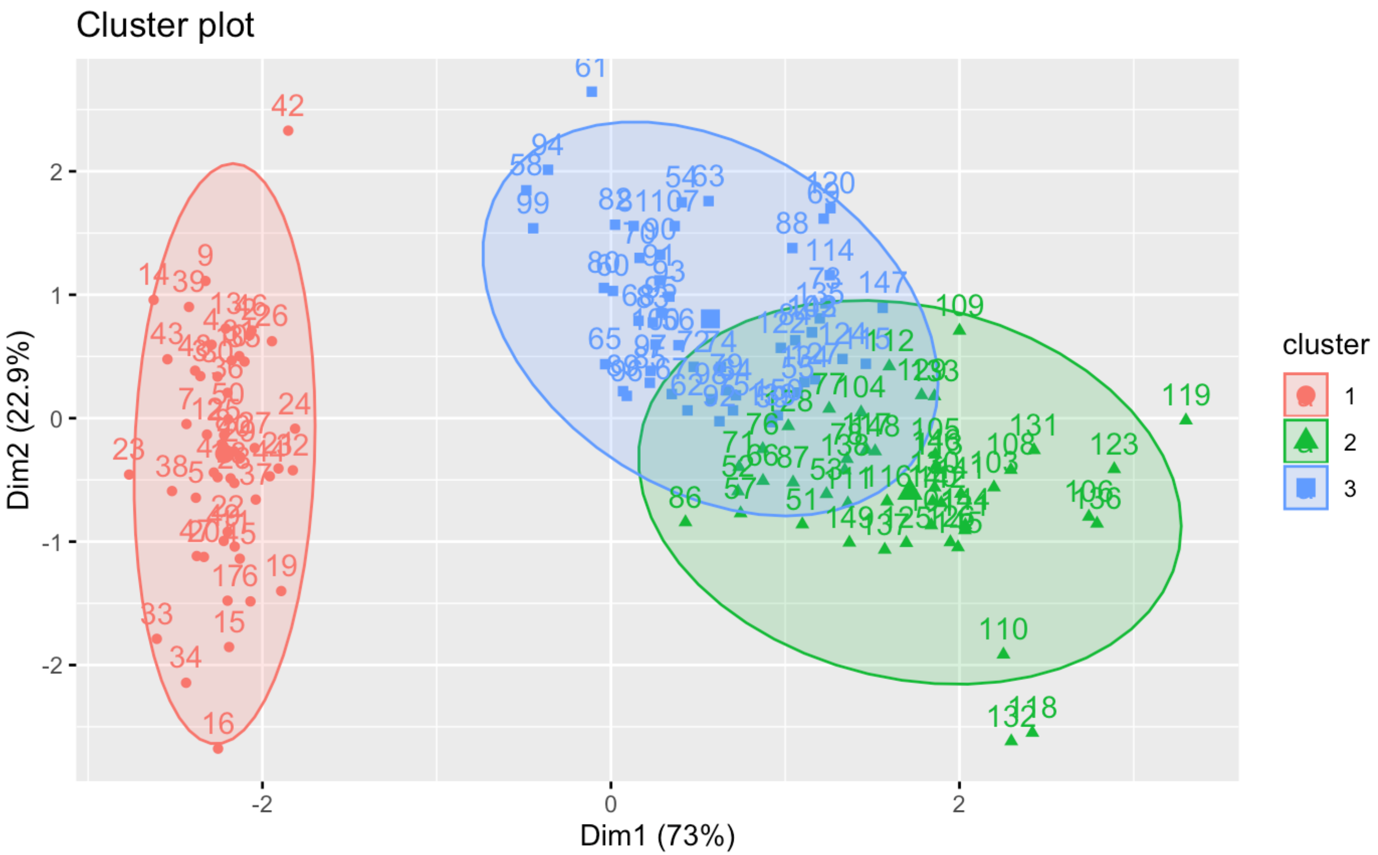


K-MEANS CLUSTERING

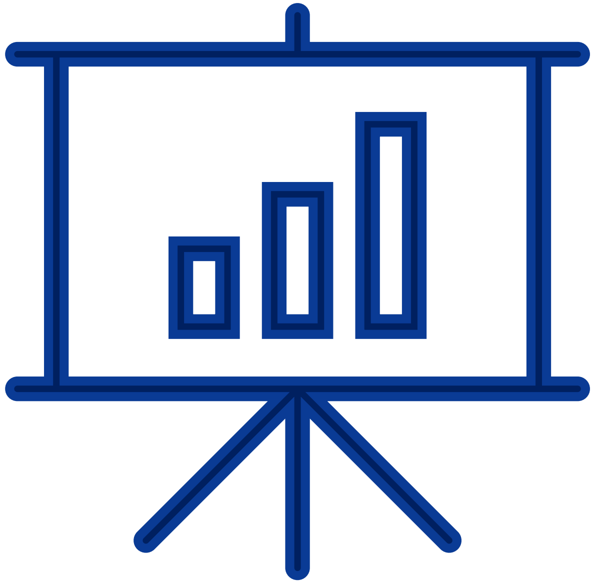
K-MEANS WITH R

fviz_cluster function (*factoextra* package) allow to visualize the clusters

Output :
ggplot object

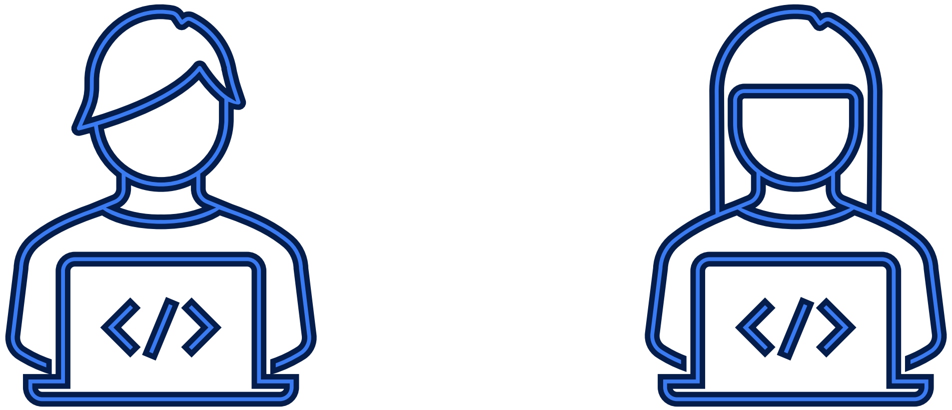


K-MEANS CLUSTERING



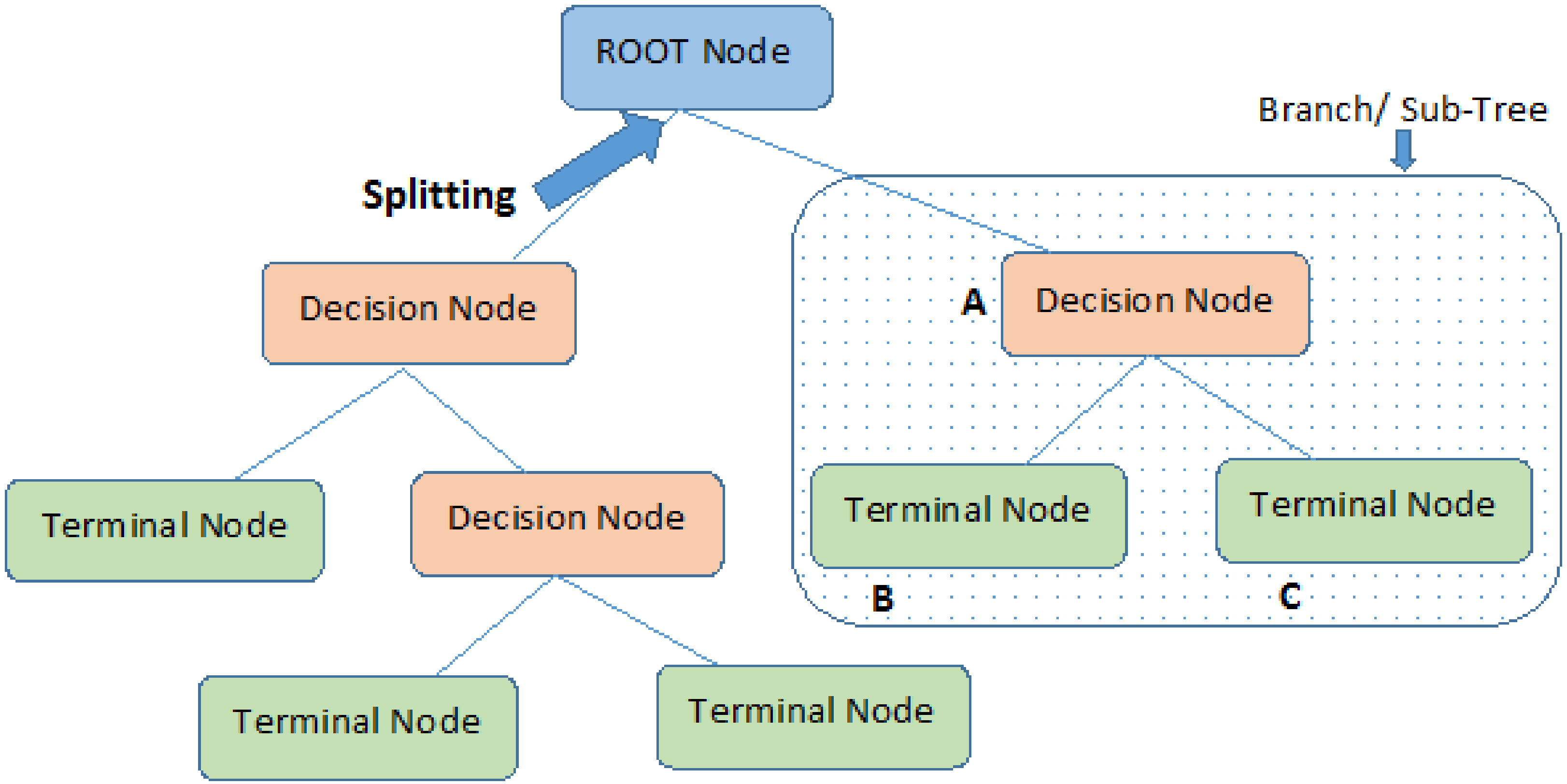
Live demo

K-MEANS CLUSTERING



Time to play !
(15 minutes)

DECISION TREES
INTRODUCTION



Note:- A is parent node of B and C.

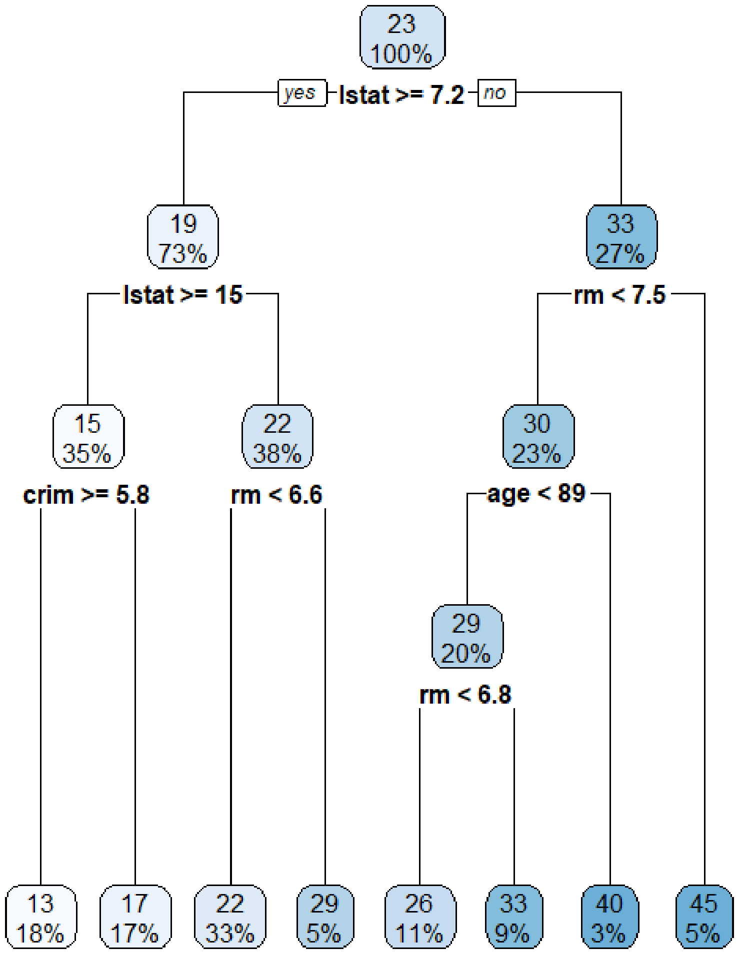
DECISION TREES
OVERVIEW

- Classification and Regression Tree (CART) algorithm invented by **Leo BREIMAN** and **Charles Joel STONE** in 1972
- Goal : find cut-off thresholds in continuous and / or categorical variables to **classify** individuals into defined groups or **predict** a continuous outcome.
- Pro : **Easy to interpret** and **visualize**
- Pro : **No need to normalize data**
- Cons : But **can overfit** the data when trees are too deep : need to **prune** them
- Cons : **Sensitive** to **small changes in data**

DECISION TREES
PRUNING TREES

Why don’t use the deepest tree ?

1. Prevents **Overfitting**
2. Improves **Generalization**
3. Reduces Model **Complexity**
4. Enhances **Interpretability**
5. Speeds Up **Training** and **Inference**
6. Facilitates Model **Maintenance**



DECISION TREES WITH R

rpart function (*rpart* package):

Parameters :

data = dataset with continuous variables
formula = outcome ~ predictors
subset = subset of data to use
method = « *anova* » for continuous outcome vs « *class* »
cp = Complexity parameter (CP) of final tree
na.action = handling of missing values

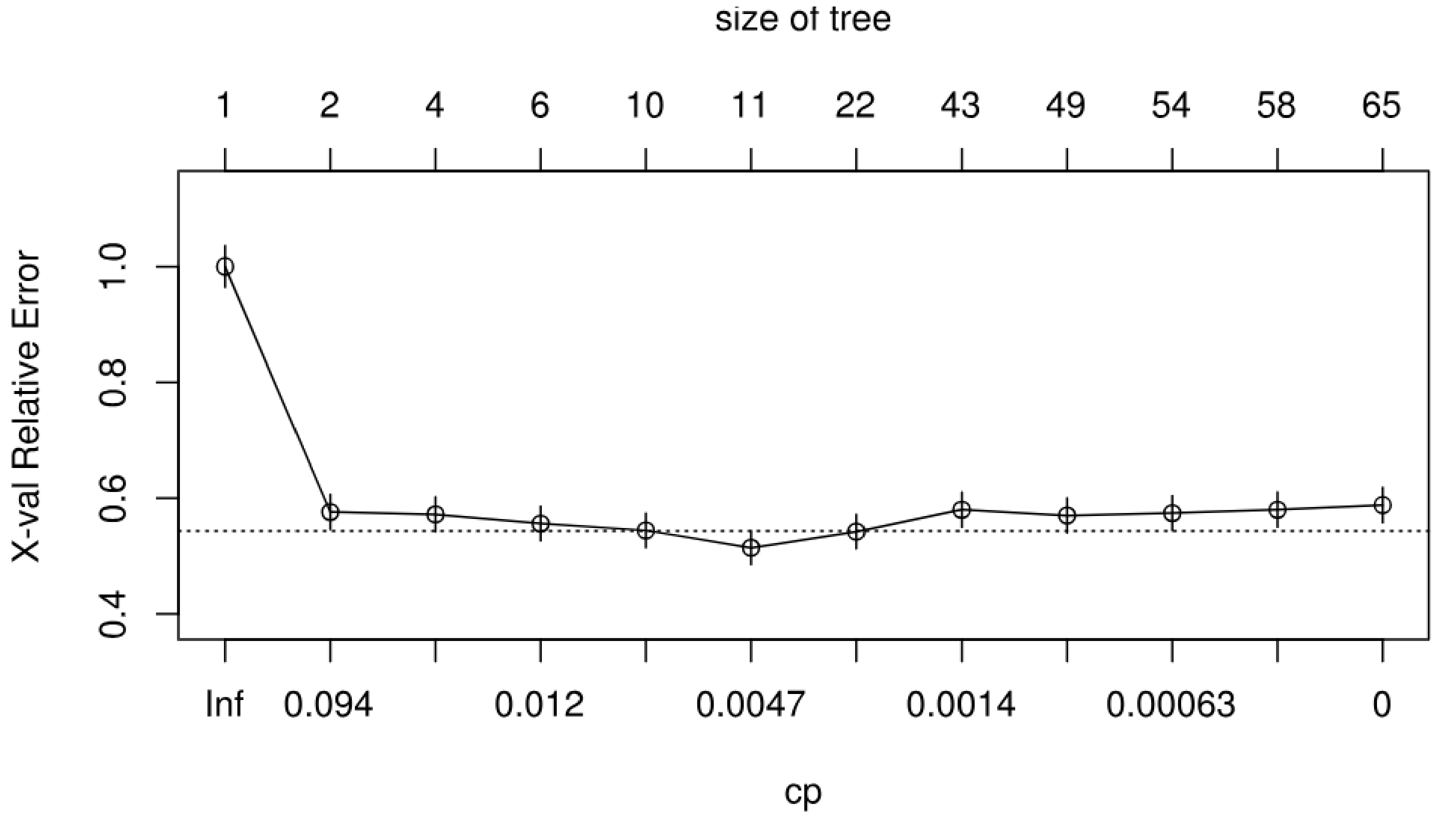
Output :

Variables selected in the final tree
Iteration information (CP, errors, standard-error)
Root node error

DECISION TREES

DECISION TREES WITH R

plotcp function (*rpart* package): allows to visualize **evolution of complexity parameter** across iterations and choose the **best tree**



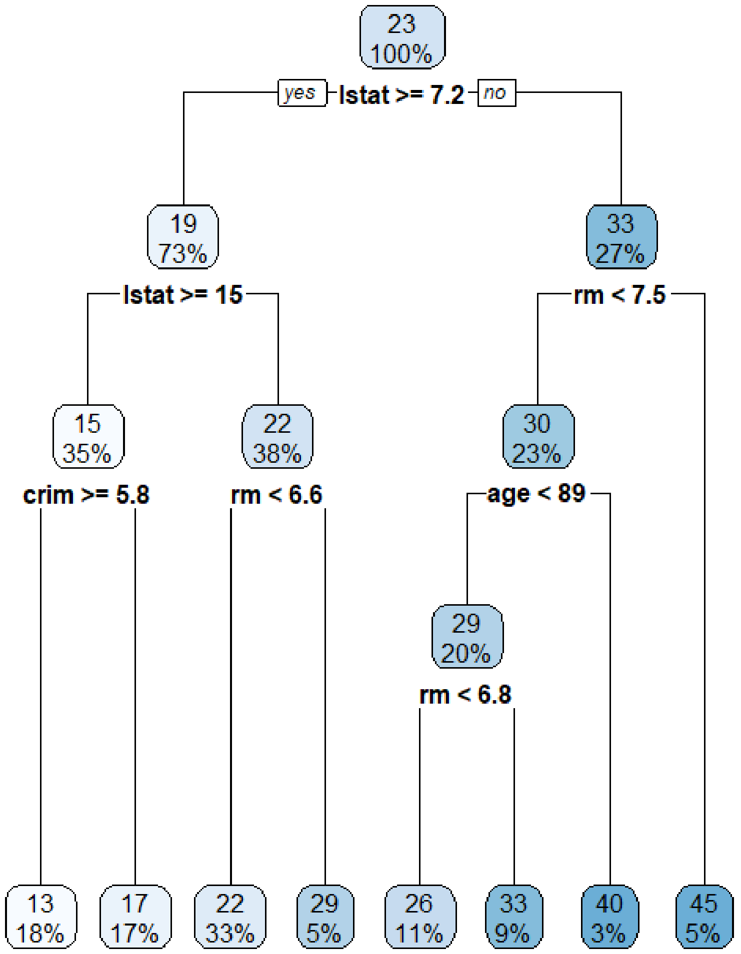
DECISION TREES

DECISION TREES WITH R

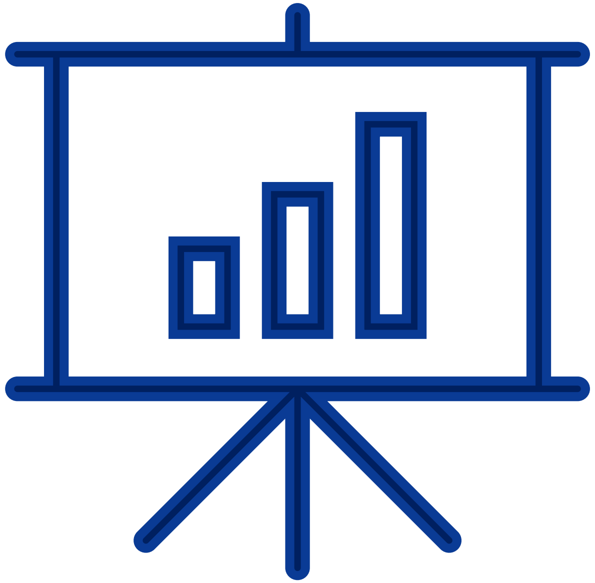
rpart.plot function (*rpart.plot* package): allows to visualize **the decision tree**

rpart.rules function (*rpart* package): summarizes the rules found by the algorithm

vip function (*vip* package): variable importance in the tree

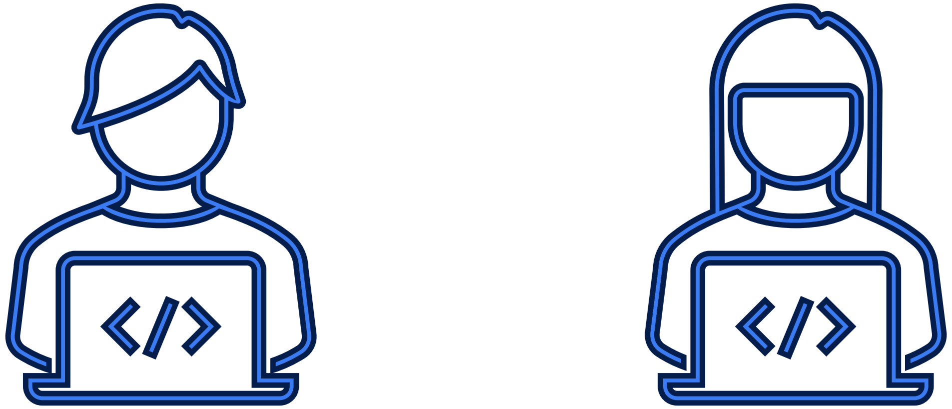


DECISION TREES



Live demo

DECISION TREES



Time to play !
(20 minutes)

QUESTIONS

05

THANK YOU FOR YOUR ATTENTION

SEPTEMBER 2025