# Unsupervised Learning

*Fabrice Jimenez*
*Data Scientist & Artificial Intelligence Engineer*

# Supervised vs Unsupervised learning

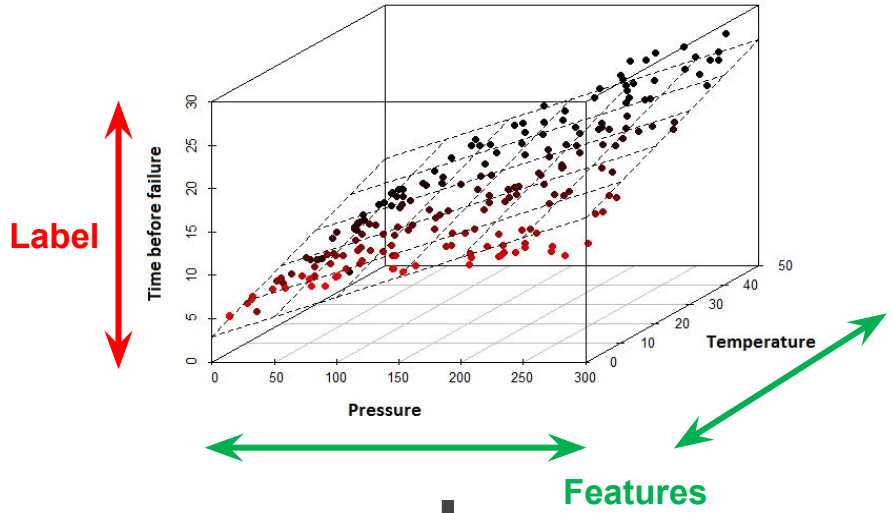
**Supervised Learning**

*You saw an example with neural networks*

Label Features

| S(ex) | H(eight) (m) | W(eight) (kg) | F(oot size) (cm) |
|---|---|---|---|
| M | 1.82 | 82 | 30 |
| M | 1.80 | 86 | 28 |
| M | 1.70 | 77 | 30 |
| M | 1.80 | 75 | 25 |
| F | 1.52 | 45 | 15 |
| F | 1.65 | 68 | 20 |
| F | 1.68 | 59 | 18 |
| F | 1.75 | 68 | 23 |

Is $(1.81, 59, 21)$ male of female?




**Classification**

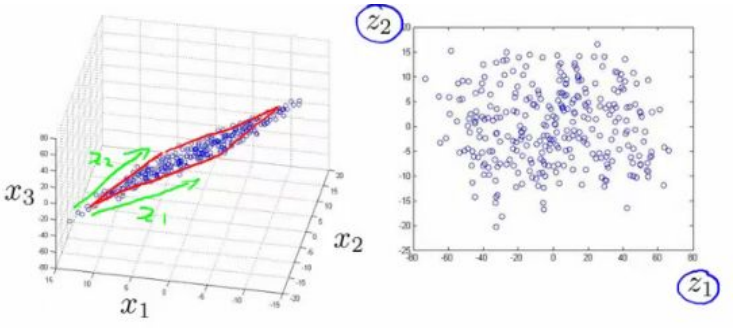
$Y = f(X)$

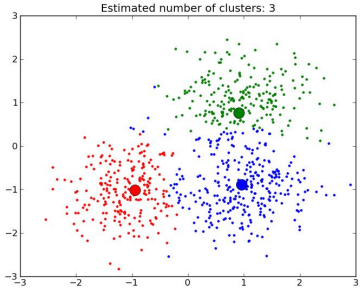
**Regression**

# Supervised vs Unsupervised learning

**Unsupervised Learning**

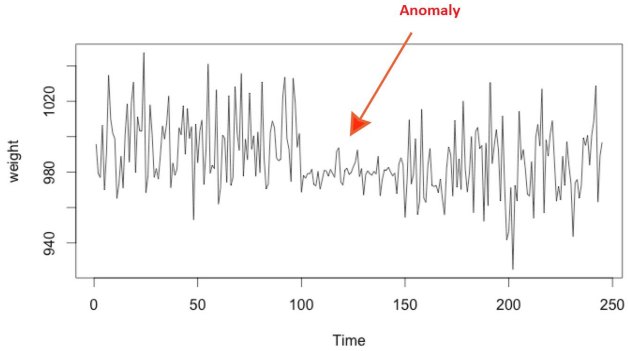
*No label to learn from: identify patterns in the data*




**Dimensionality reduction**

**Clustering**




**Anomaly detection**

# A “real-life” time series use case

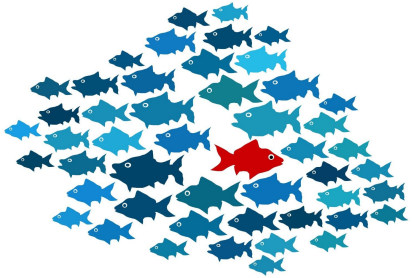
# The dataset

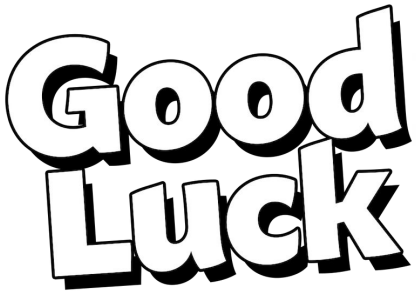
Aircraft systems are recording values of parameters such as speed, temperature, pressure, electrical current values...

**4 parameters recorded:**
**p1, p2, p3, p4**

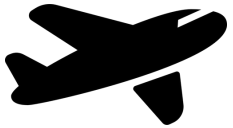
# The question

An aircraft system expert comes to see you (data scientist) with this dataset, and asks you to:
**“Build an algorithm to detect cycles that are abnormal.”**






# Time series use case follow-up



**Formulating the problem!**

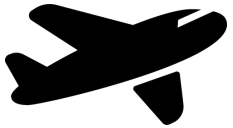
What inputs do I have?

- ➔ Only parameters, no quality indicator, no operational context…
- ➔ No labels giving examples of anomalies…

What can I do?

- ➔ No labels: **unsupervised learning**
- ➔ No definition of what is abnormal: study behavior of parameters to **identify patterns different from the majority**

# Time series use case follow-up



**Formulating the problem!** → **Formulating validation strategy**

How will I know if my results are acceptable or not?

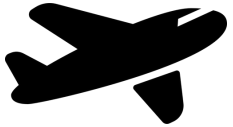
➔ No example of abnormal cycles to build confusion matrix: **no strong validation possible**

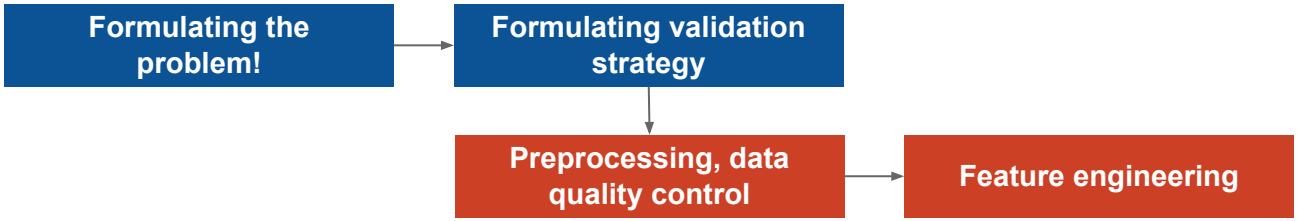
We can use **weak validation**: present abnormal cycles to a business engineer

➔ they will tell if detected patterns are really abnormal from operational point of view

➔ **several iterations from there**

# Time series use case follow-up





# Your turn!

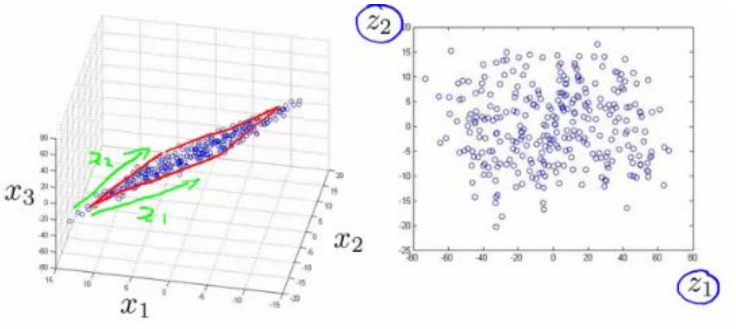
Let’s explore our dataset, and prepare it for our task with Python...

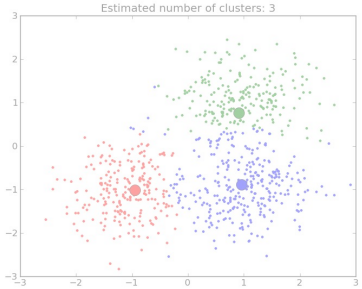
# From correlation to dimensionality reduction

# Supervised vs Unsupervised learning

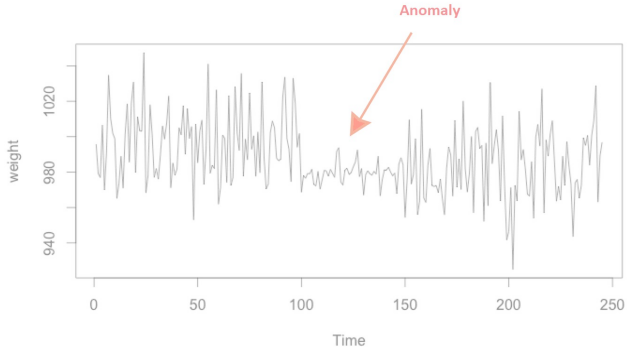
**Unsupervised Learning**

*No label to learn from: identify patterns in the data*




**Dimensionality reduction**

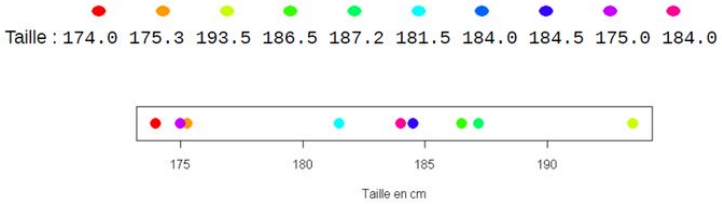



**Clustering**

**Anomaly detection**

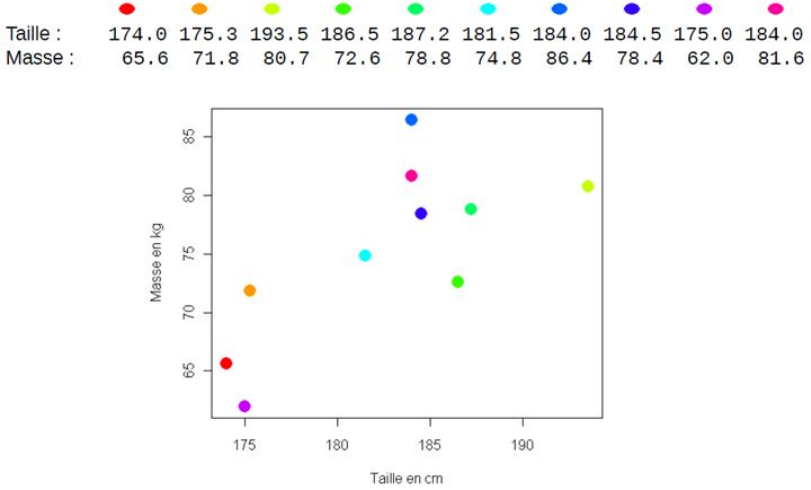
# Statistical Indicators in 1D vs 2D

**1D - With one single variable: OK**




Mean, median, standard deviation, variance, quantiles...

**2D - With 2 variables: how can we describe the behavior of one of them with respect to the other?**

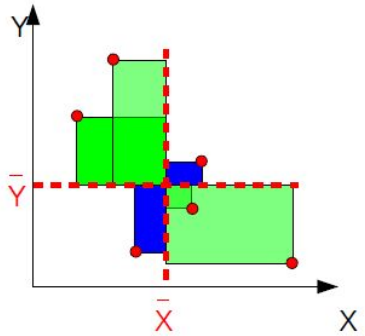


# Statistical Indicators in 2D: linear correlation

**Covariance:** mean of the product of differences to the means
*Generalization of 1D variance: var(X) = cov(X,X)*

$$\mathrm{cov}(X,Y)=\frac{1}{n}\sum_{i=1}^{n}\left(X_i-\bar{X}\right)\left(Y_i-\bar{Y}\right)$$

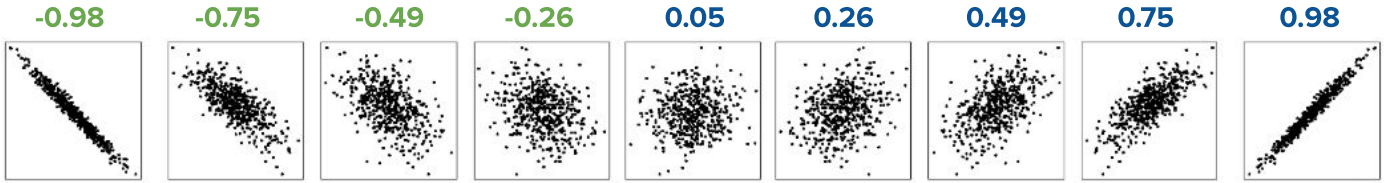
Be careful: just like variance, covariance depends on the unit = product of units of the variables!




**Pearson linear correlation coefficient:** we normalize the covariance

$$\rho(X,Y)=\frac{cov(X,Y)}{(\sigma_X\sigma_Y)}$$

$\rho(X,Y)$

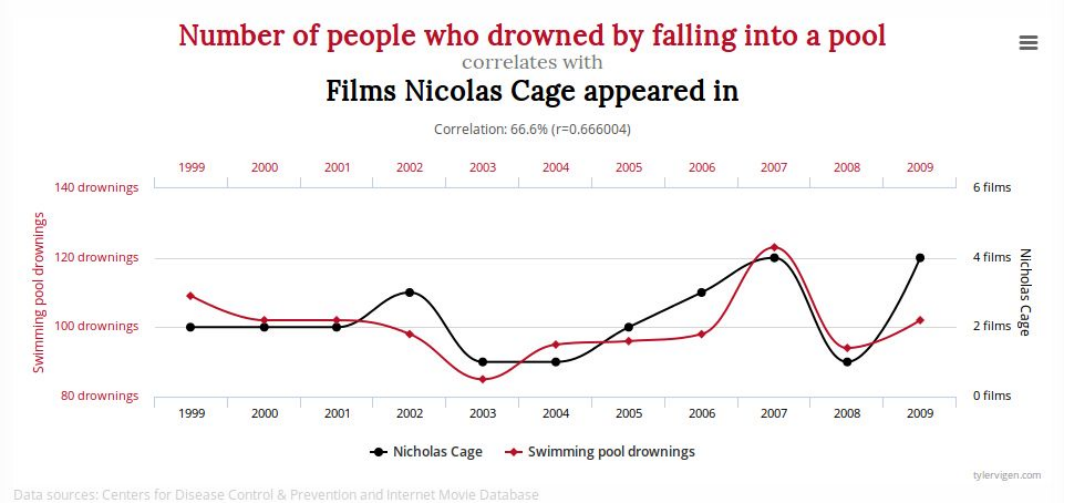



**Describes LINEAR relationship between variables**

# Statistical Indicators in 2D: interpretation!

**Be careful of quick interpretations!**
**Correlation ≠ Causality!**

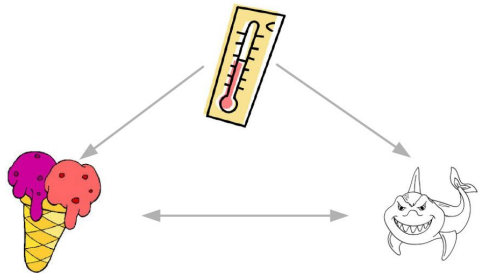
**<u>Example 1:</u> factual correlations without link**




**<u>Example 2:</u> third factor**

**The frequency of shark attacks is strongly correlated to the sales of ice cream! Does eating ice cream make us more attractive in the eyes of sharks?**

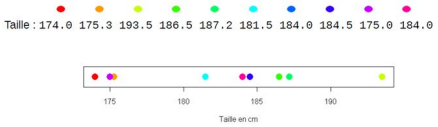
**Situation of parallelism with a third factor behind: the high temperatures!**



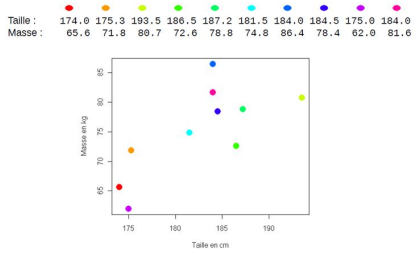
Think about those examples when you interpret results…

# Next: Statistical Indicators in 1D vs 2D vs nD

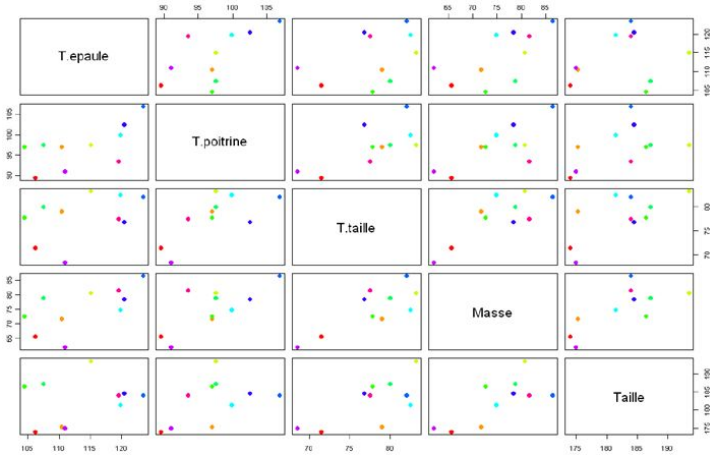
**1D - With one single variable: OK**




**2D - With 2 variables: OK**



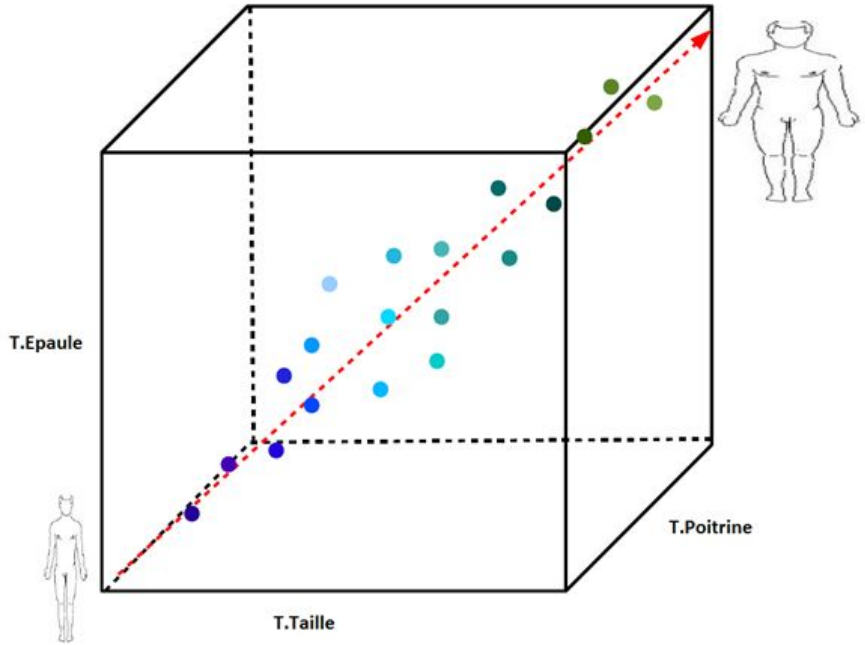

**nD - With n variables???**



# Dimensionality reduction

**Example with 3 dimensions:** the concept of correlation generalizes to groups of variables

➝ *redundant information, do we really need 3 dimensions to describe the dataset?*



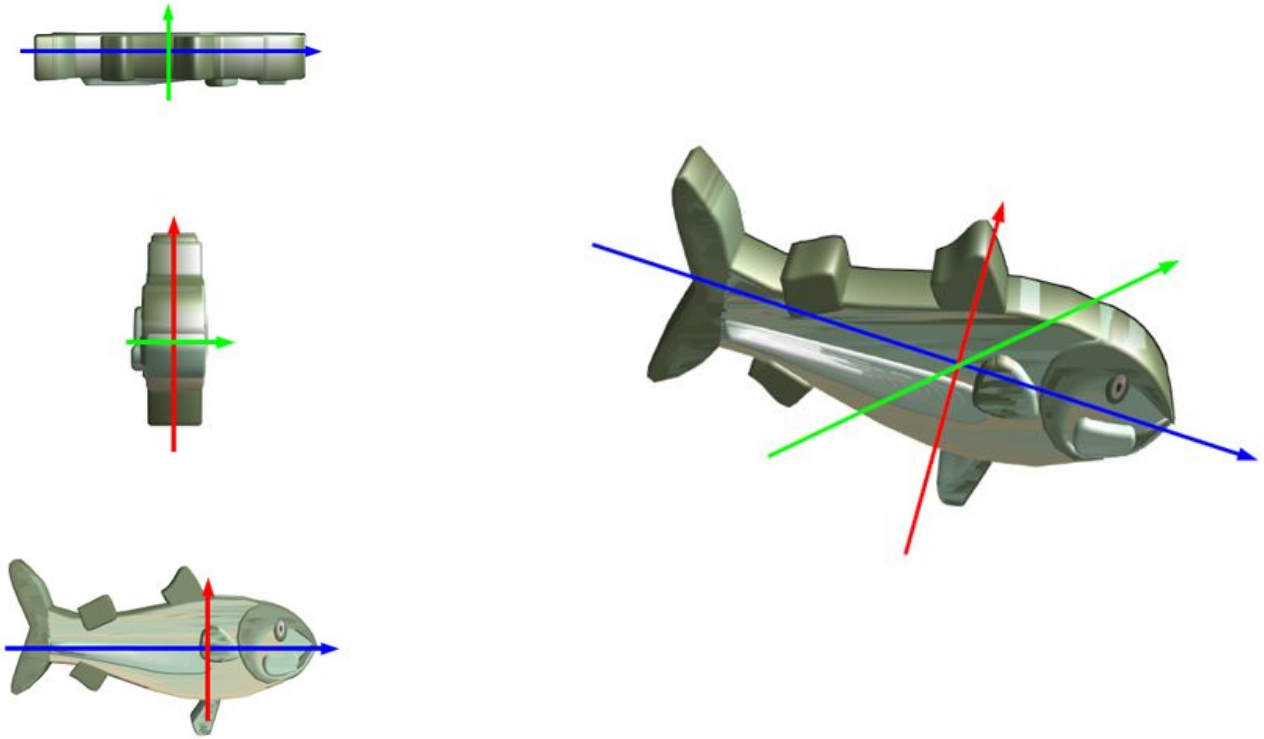

**Introduction of a component describing the common evolution of T. Taille (waist size), T. Poitrine (chest size) et T. Epaule (shoulder size)**

➝ *We want to keep a maximum of information with a minimum of variables*

**New component: can be seen as “stature”**

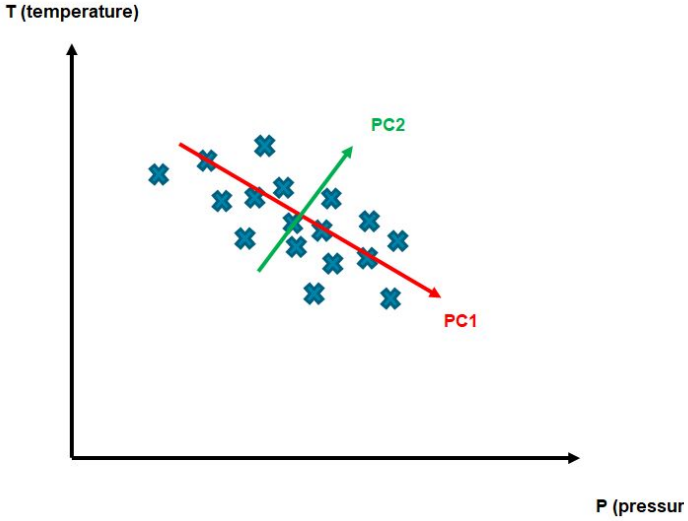
# Principal Components Analysis: intuition

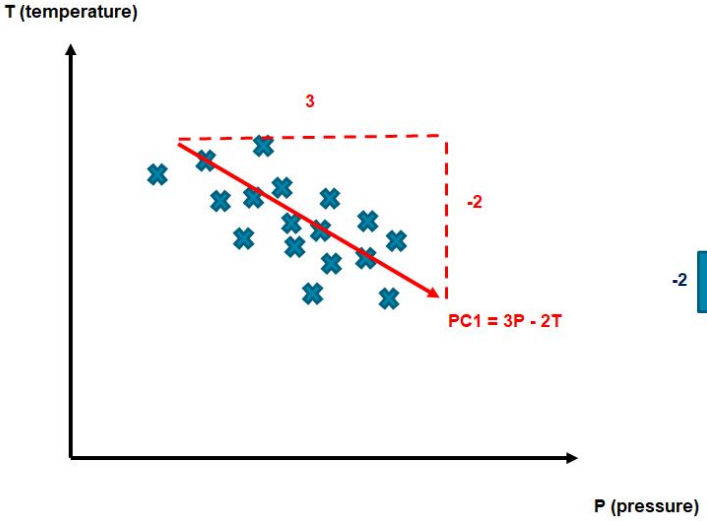
**It is looking for a rotation of the initial variables, an angle of view, in which the projected point cloud is the most spread.**

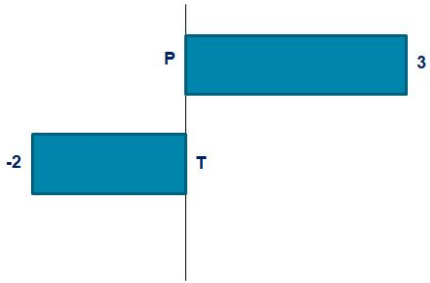


# Principal Components Analysis (PCA)

**Particular technique of dimensionality reduction:** linear combination of variables, to maximize the variance of the data



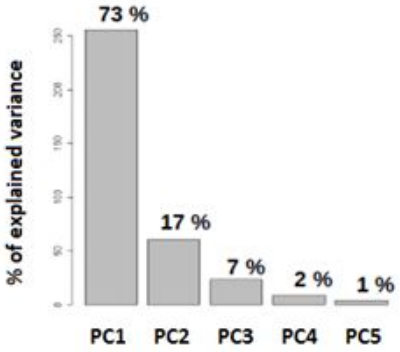




**Usual practice:** keep the n first principal components to reduce dimension

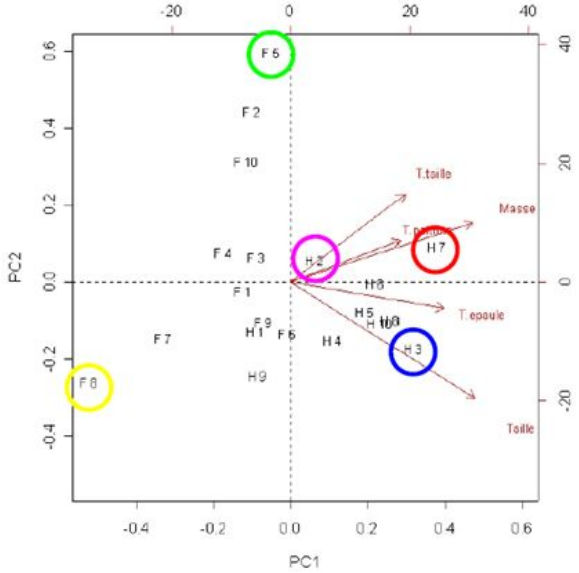
It is 1 technique among many (t-SNE, autoencoder neural networks...): whole field of machine learning !

# Principal Components Analysis: projection

**Projection: visualization of dataset points and initial variables in the basis of principal components**




Representation of explained variance

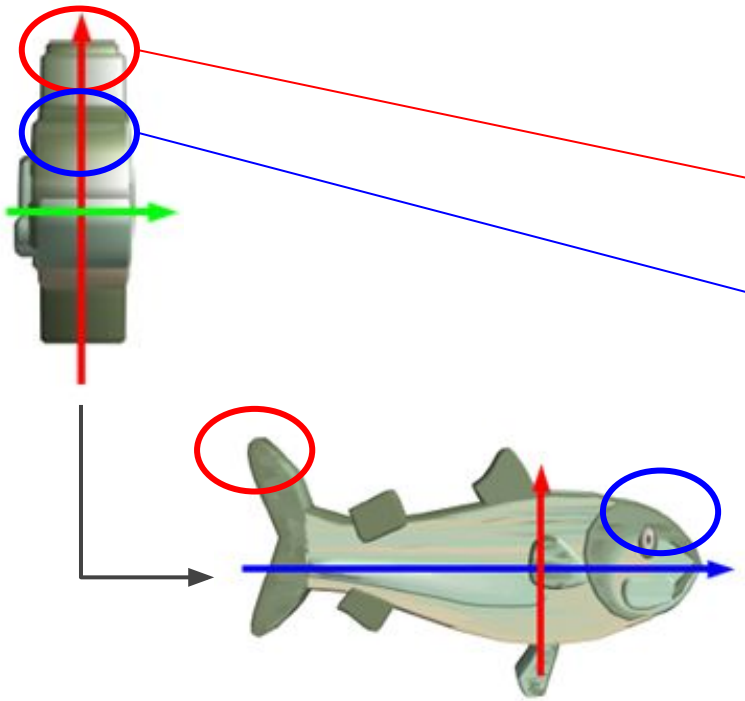


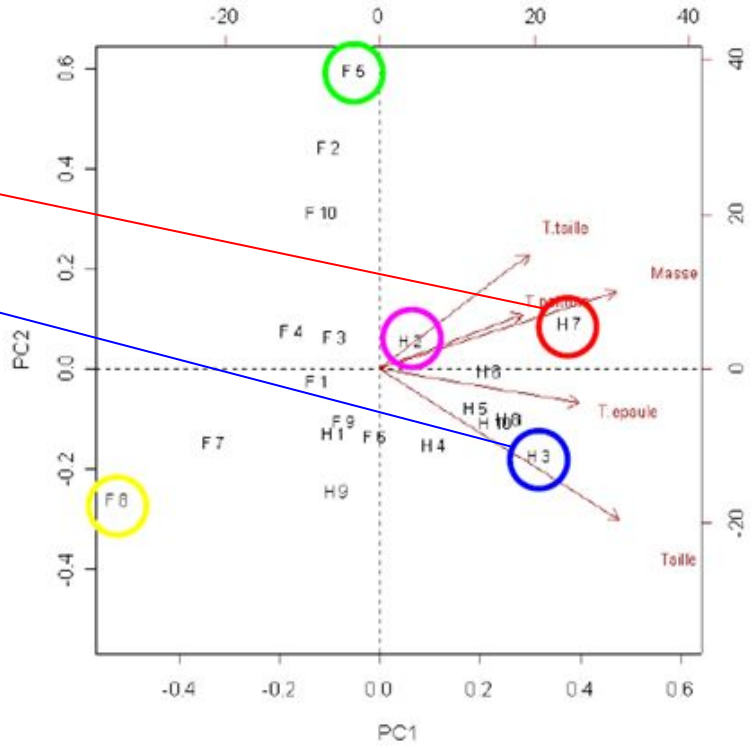

Projection for interpretation

**Correlations: groups of correlated initial variables are projected in vectors with similar directions (small angle)**

# Principal Components Analysis: projection

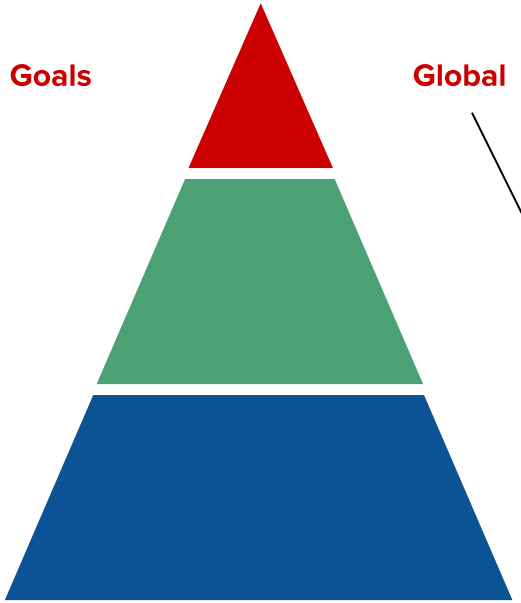
**Projection: visualization of dataset points and initial variables in the basis of principal components**






**Be careful with the projection's interpretation !**

# What do I have to remember?



Goals

Global

Means

Techniques

Detail

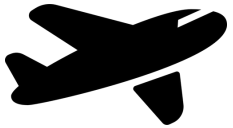
Visualize patterns with many variables, identify links (not causality!)

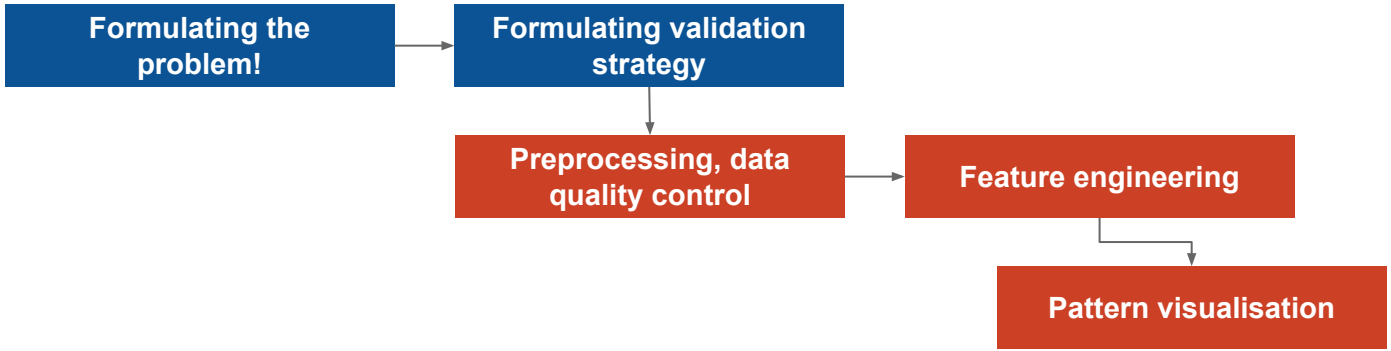
Correlation, dimensionality reduction, careful interpretation

Pearson linear correlation, Principal Components Analysis, Projection plots

Focus here

# Time series use case follow-up





# Your turn!

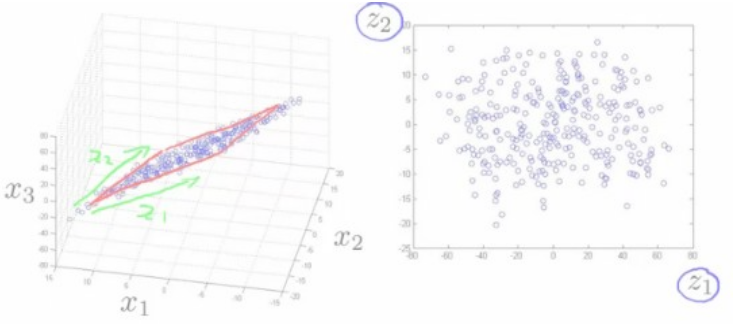
It’s time to apply all of that together
with a bit of Python...

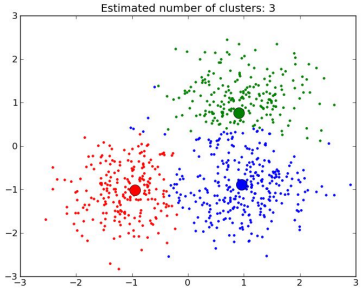
# Clustering

# Supervised vs Unsupervised learning

**Unsupervised Learning**

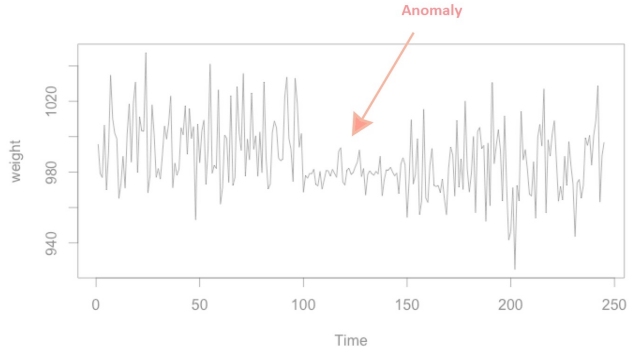
*No label to learn from: identify patterns in the data*




**Clustering**

Dimensionality reduction



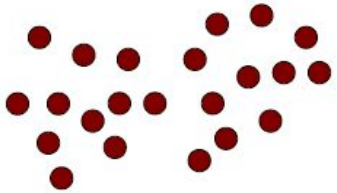

Anomaly detection

# Clustering: illustration with K-Means

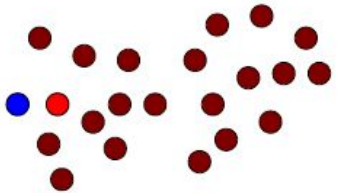
**Methods to detect groups of similarity in a dataset**
**Many techniques can be used. One of them is popular, computationally cheap, and easy to understand: K-Means**

**Prerequisites: distance measure between individuals, predefined number of groups**

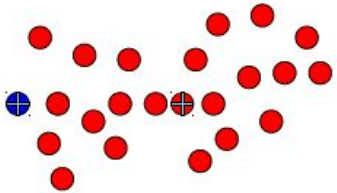
**1/ Initial dataset**



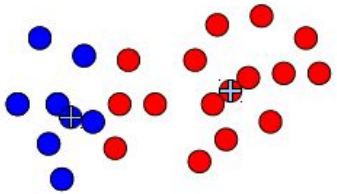
**2/ Random initial centers (n=2)**



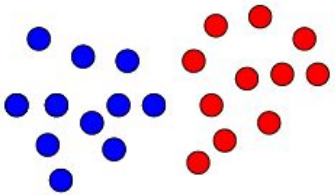
**3/ Groups = closest points to each center. Compute new centers**



**4/ Groups = closest points to each center. Compute new centers**



**5/ Stop when stabilised**



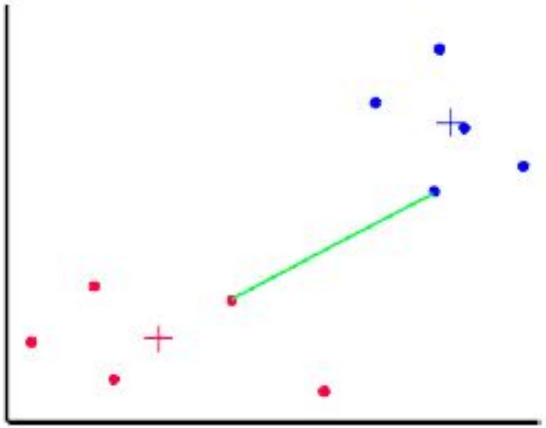
# Clustering: illustration with HAC

**Hierarchical Ascendant Clustering: different technique with different properties**

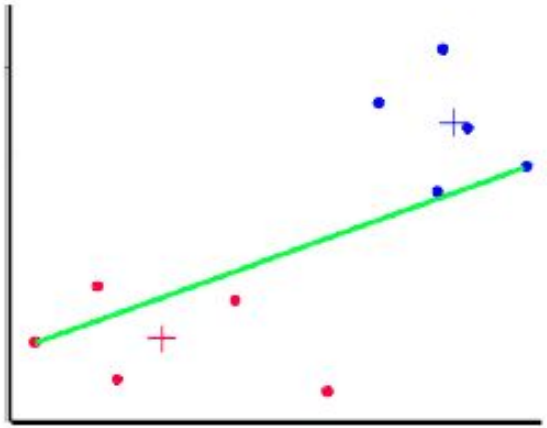
**Prerequisites: distance measure between individuals, distance measure between groups**

**Examples of distance measures between groups:**

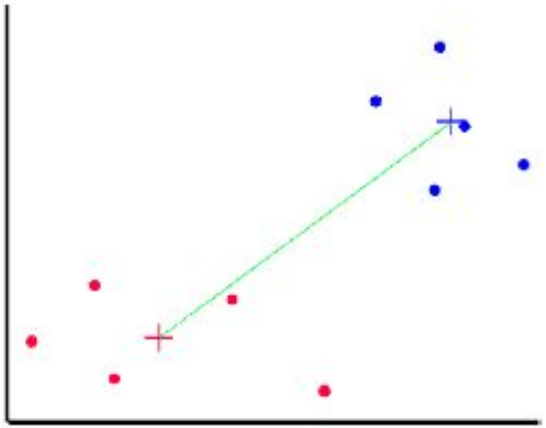
**Minimum**



**Maximum**



**Average**

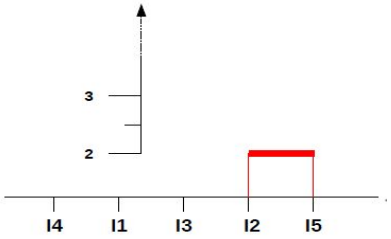


# Clustering: illustration with HAC

**Toy example: 5 individuals, 3 variables**

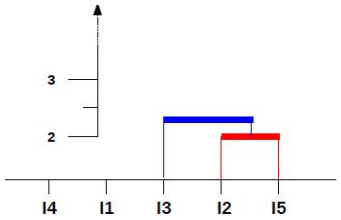
| | V1 | V2 | V3 |
|---|---|---|---|
| I1 | 1 | 2 | 3 |
| I2 | 4 | 2 | 5 |
| I3 | 4 | 3 | 7 |
| I4 | 8 | 9 | 6 |
| I5 | 4 | 2 | 3 |

Step 0: Compute distances between individuals, group the 2 closest into N1

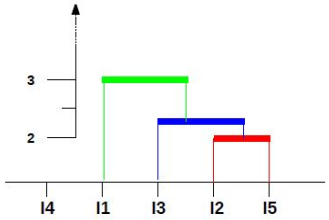



| | I1 | I2 | I3 | I4 |
|---|---|---|---|---|
| I2 | 3.61 | | | |
| I3 | 5.10 | 2.24 | | |
| I4 | 10.34 | 8.12 | 7.28 | |
| I5 | 3.00 | 2.00 | 4.12 | 8.60 |

Iterations: Compute distances between individuals / groups, group the 2 closest into Nx

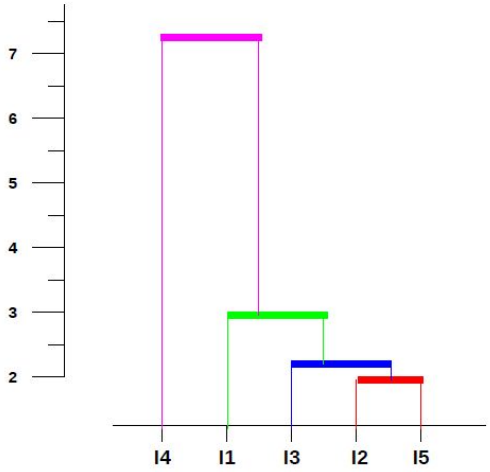



| | I1 | I3 | I4 |
|---|---|---|---|
| I3 | 5.10 | | |
| I4 | 10.34 | 7.28 | |
| N1 | 3.00 | 2.24 | 8.12 |




| | I1 | I4 |
|---|---|---|
| I4 | 10.34 | |
| N2 | 3.00 | 7.28 |

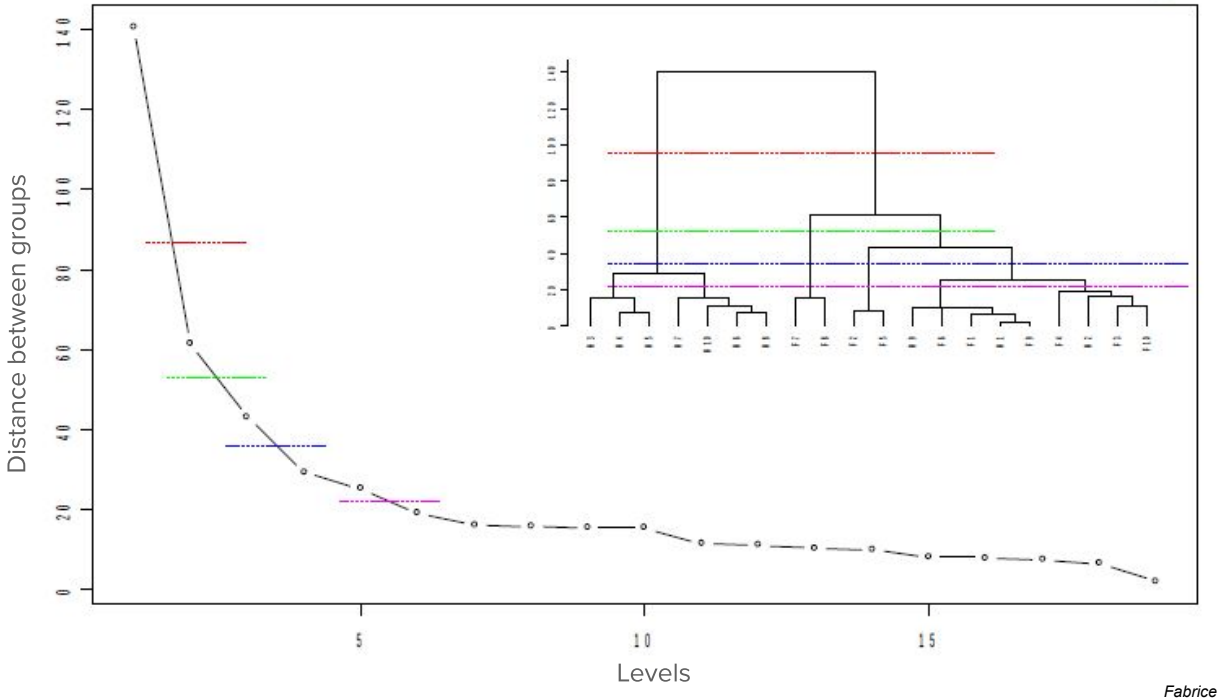
End: complete Dendrogram is built



# Clustering: illustration with HAC

**Dendrogram = not directly the groups but a decision tool**

**Cut at the right level to obtain groups: always criteria to define!**



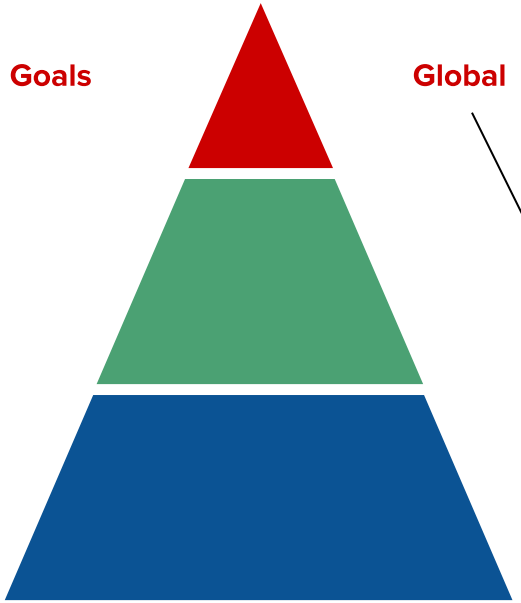
# Clustering: illustration with HAC

| | PROS | CONS |
|---|---|---|
| K-Means | Easy process<br>Low computing cost | Not deterministic (initial centers)<br>Needs number of groups |
| HAC | No need for number of groups<br>Deterministic process | High computing cost<br>Distance measure between groups to define |

Typical illustration of tradeoff to make... **Hands-on experience brings best practices**, for example:

- If highly variable K-means: perform HAC, identify number of groups and centers to initialize K-means
- If too many data points for HAC: perform K-means with high number of groups, then HAC on the centers
- ...

# What do I have to remember?



Goals

Global

Build groups of similarity

Means

Clustering, evaluate tradeoff criteria between methods

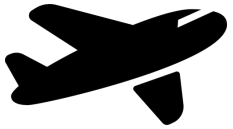
Techniques

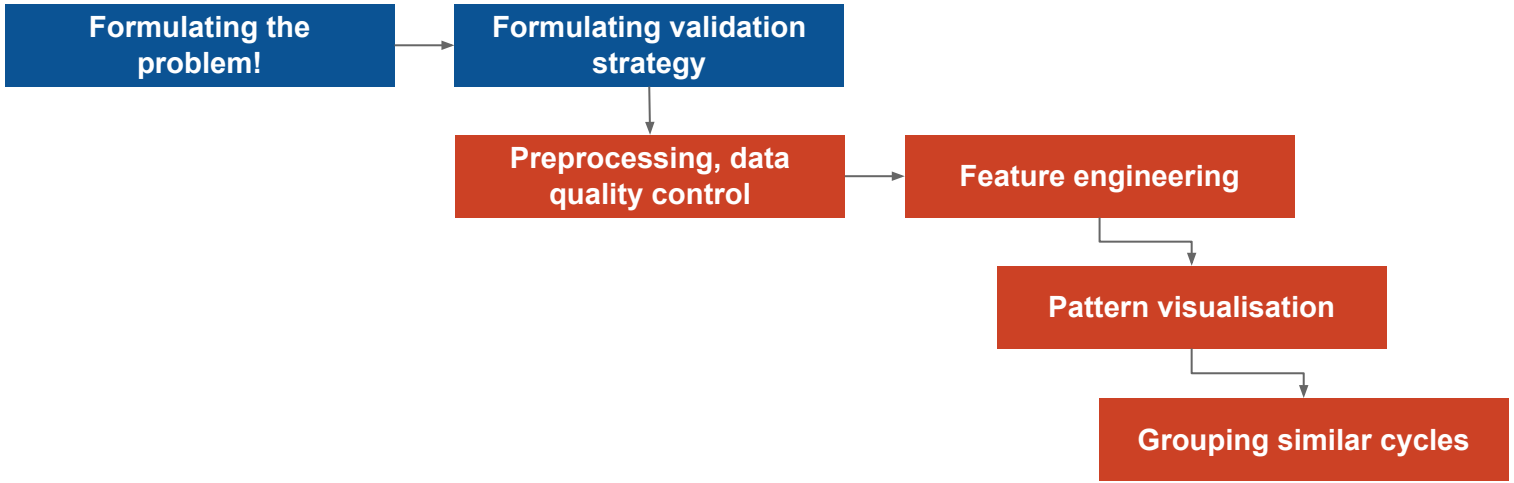
Detail

K-Means
Hierarchical Ascendant Clustering
Dendrogram

Focus here

# Time series use case follow-up





# Your turn!

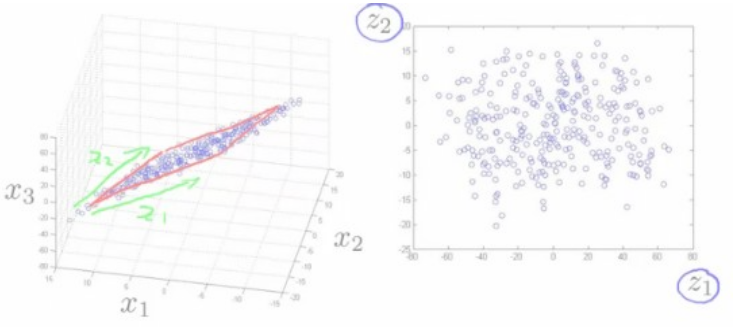
It’s time to apply all of that together
with a bit of Python...

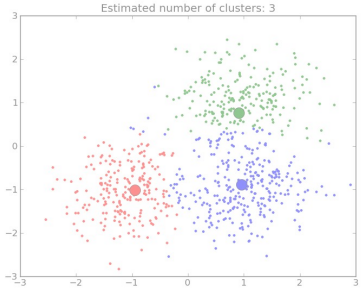
# Anomaly detection

# Supervised vs Unsupervised learning

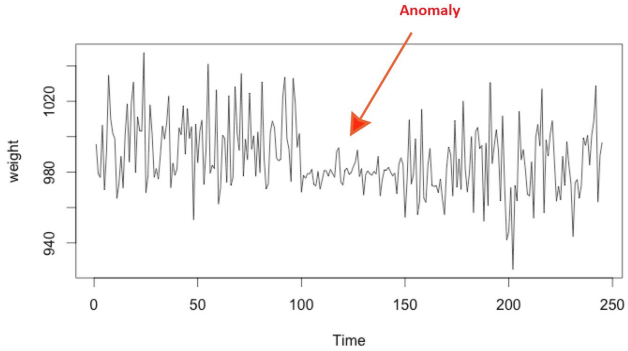
**Unsupervised Learning**

*No label to learn from: identify patterns in the data*




Dimensionality reduction




Clustering

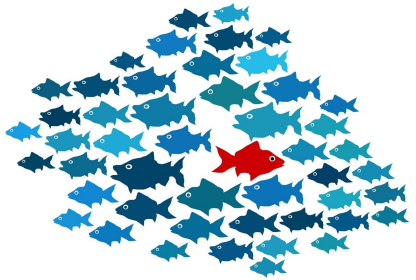
**Anomaly detection**

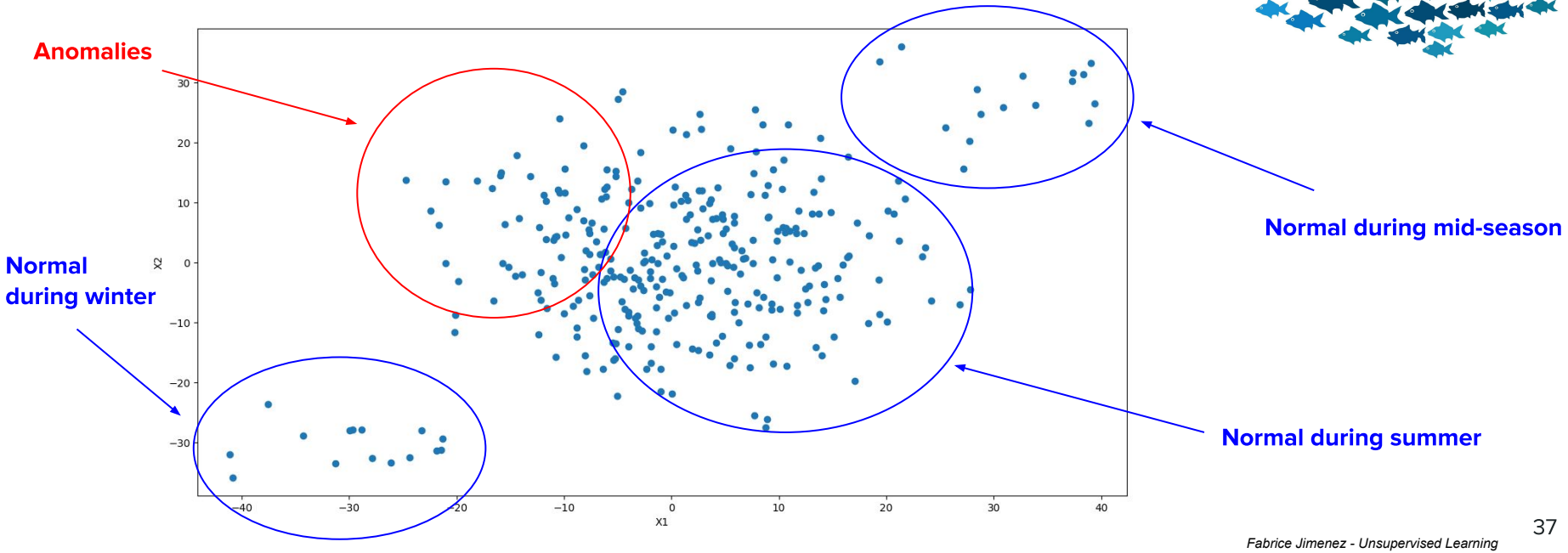
# Anomaly detection: definition and scope

**What is an anomaly?**

**1/ Generally: a rare individual (row) in a dataset that differs significantly from the majority of the data**

**2/ Sometimes:** anomalies are not so rare, and may not be so different from the majority of the data...





# Anomaly detection: definition and scope

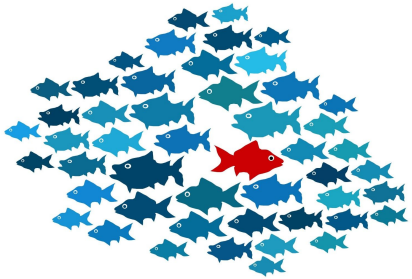
**Why not using Supervised Learning with labeled dataset?**

**Very unbalanced dataset**

*5 anomalies given 100 000 normal points...*

**Lack of coverage of all anomaly types**

*Anomaly = something not expected, what if a new type happens...*



**We need other approaches...**

**Outlier detection:** the dataset contains anomalies in the sense of statement 1/ (rare + statistically different)

→ Detect elements in this same dataset which differ from the majority of the data

**Novelty detection:** you have a clean dataset without anomalies (in the sense of 1/ or 2/)

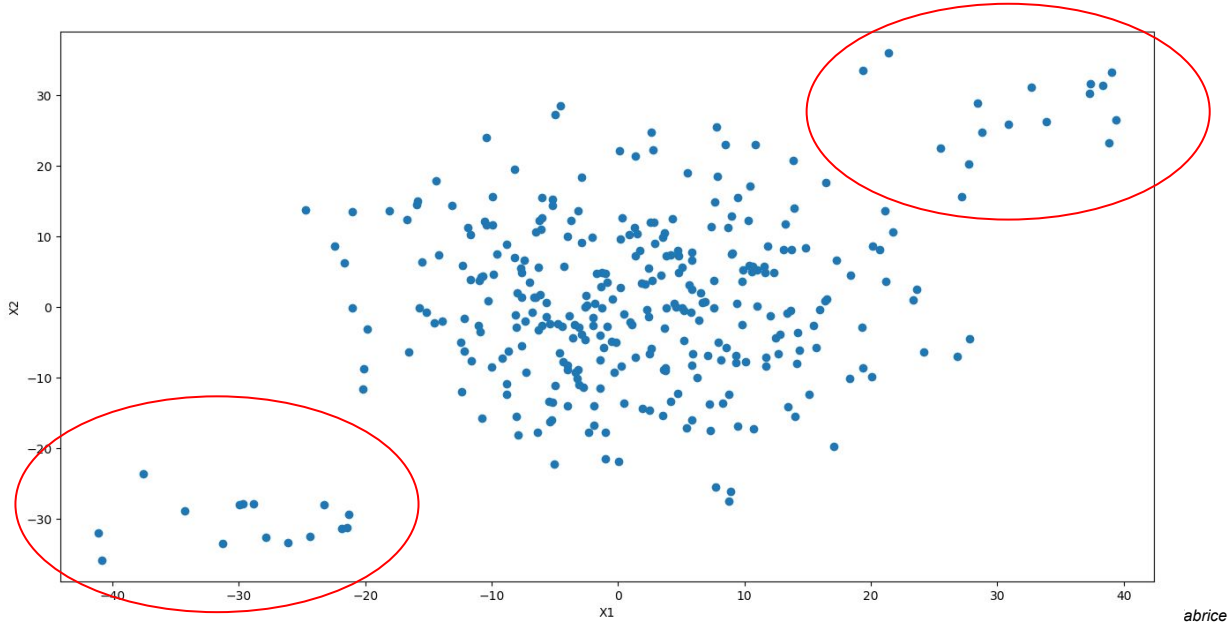
→ Learn the normal behavior, to be able to check if a new item is normal or an anomaly

→ Some techniques can be used for both, but **be aware** of the approach you are using, and why...

# Outlier detection

**1/ Anomaly = a rare individual (row) in a dataset that differs significantly from the majority of the data**

**Outlier detection:** the dataset contains anomalies in the sense of statement 1/
→ Detect elements in this same dataset which differ from the majority of the data



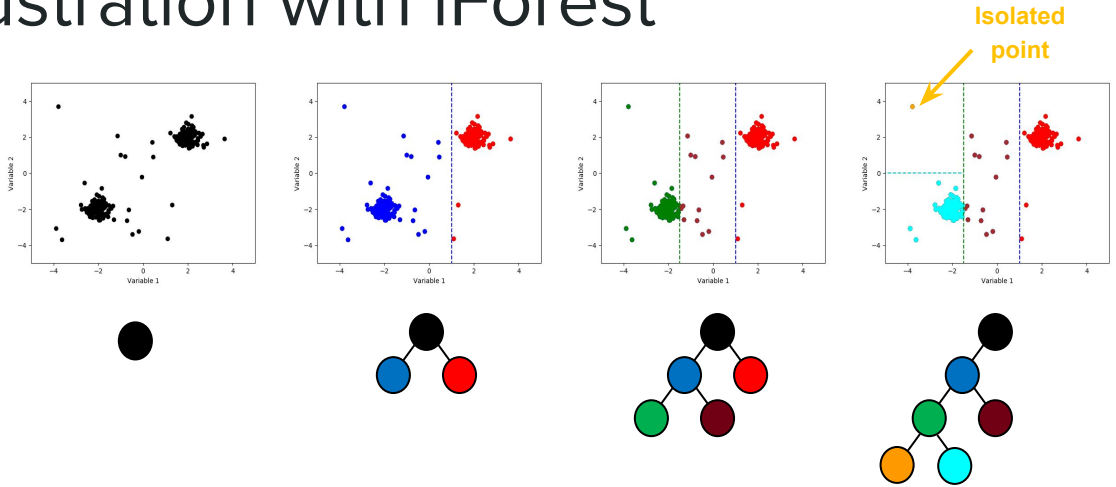
# Outlier detection: illustration with iForest

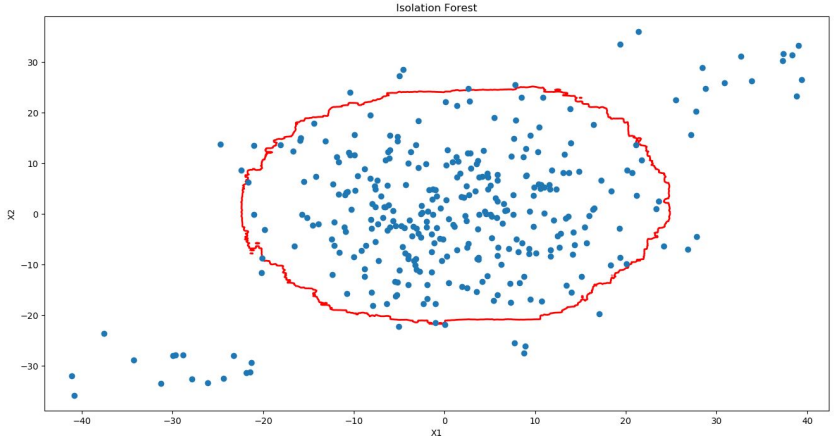
**Example of technique - Isolation Forest**

➔ 1/ Build Isolation Tree:
Split entire dataset with random variables and random thresholds

➔ 2/ Repeat with 100, 1000 trees...

➔ 3/ Average depth of a point in the forest
≔ anomaly score




**Low depth = high anomaly score**
**High depth = low anomaly score**

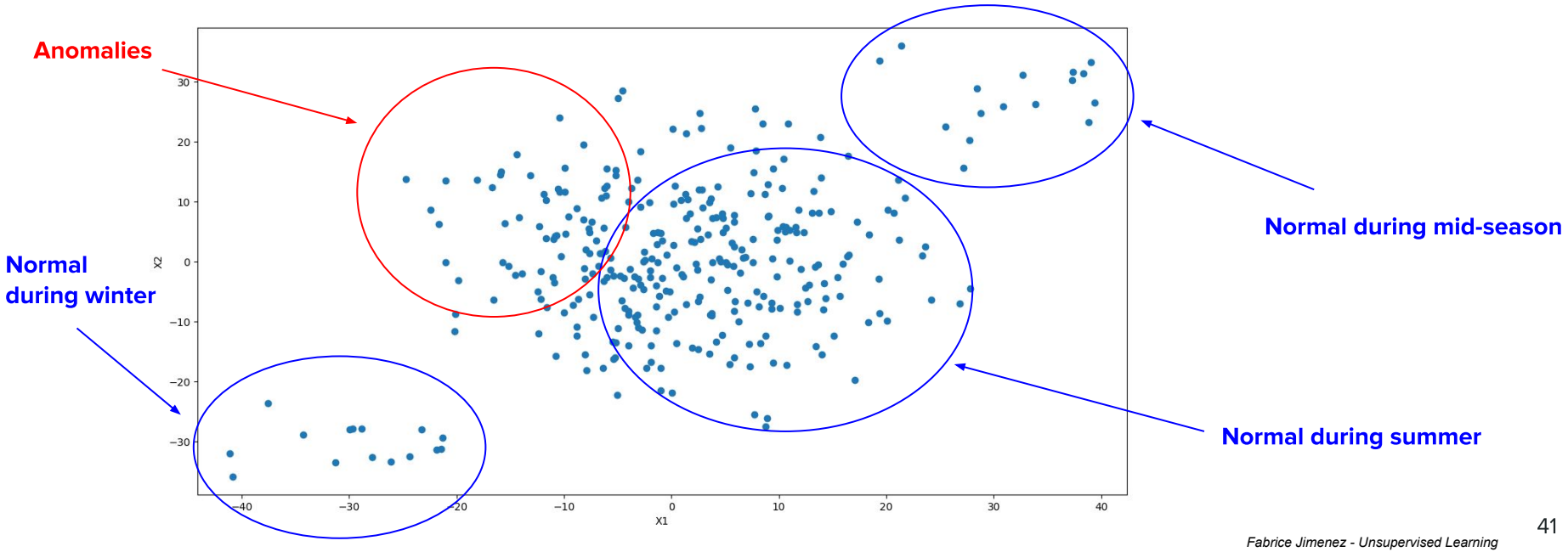
Threshold to define to be an anomaly or not!

# Novelty detection

**2/ Anomalies = not so rare**, and may not be so different from the majority of the data...

**Novelty detection:** you have a clean dataset without anomalies (in the sense of 1/ or 2/)

➔ Learn the normal behavior, to be able to check if a new item is normal or an anomaly



# Novelty detection

## Basic principle

**Clean dataset** without anomalies: **“learn” the normal behavior.**
Predict the value / score of new points to find out if they match the normal behavior or not

➝ Unsupervised methods like iForest can also be used in this case, but new possibilities!

## New possibilities

Why not using supervised learning to learn the normal behavior?

| v1 | v2 | v3 |
|---|---|---|
| 8.4 | 15 | 2.2 |
| 9.1 | 10 | 5.1 |
| ... | ... | ... |

Model 1: v1 = f(v2,v3)
Model 2: v2 = f(v1,v3)
Model 3: v3 = f(v1,v2)

**Predict each variable** by using the others as features:
➝ Neural Network
➝ Other supervised technique...

➝ A new point comes in: (x1,x2,x3)
➝ Compute the predictions [x1] = f(x2,x3), [x2] = f(x1,x3), [x3] = f(x1,x2)
➝ Compute the errors [xi] - xi : squared error, absolute error...

**High error = does not fit the “normal” model = high anomaly score**

# Outlier detection: score VS decision

**Be careful!**

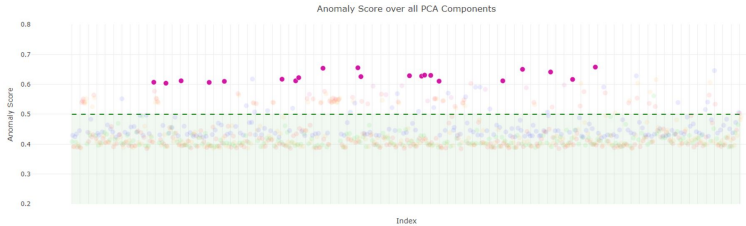
- All outlier & novelty detection methods give only a **relative measure of abnormality**
  - How different are the points compared to majority / reference data

Continuous scores

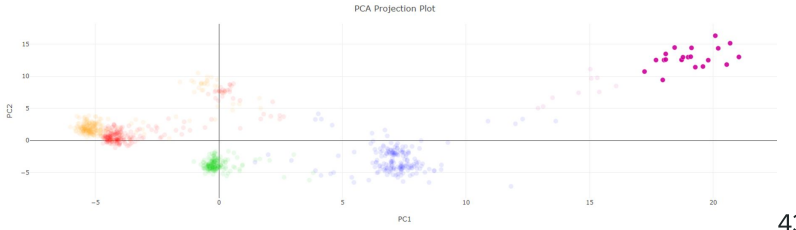
- The **decision itself (outlier or not)** is proposed by default in those methods, but it **always requires threshold tuning**!
  - Always need for human intervention, especially with complex interdependent systems!
  - For example: cross the anomaly scores with manual cluster analysis with PCA, geometrical interpretation...

NOT because technology is not mature enough...

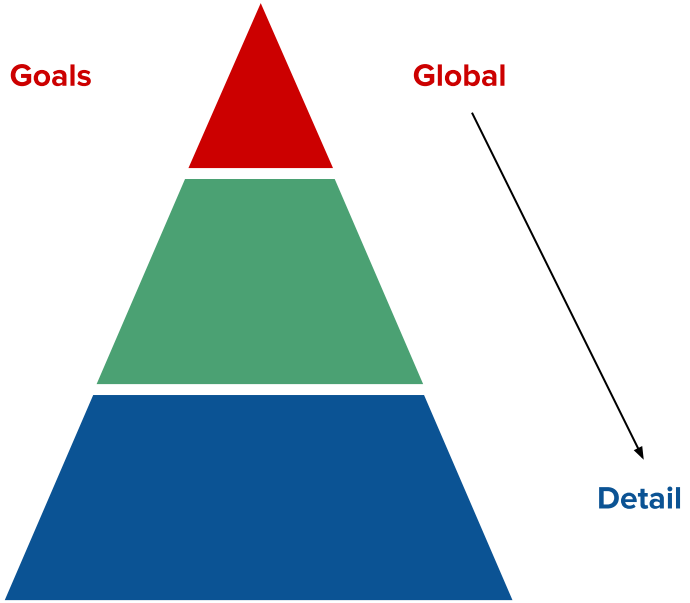
BUT **because the problem is badly formulated!**
"Anomaly" is not clearly defined a priori, and statistics will never tell you what it is!




Human intervention for threshold and / or decision!



# What do I have to remember?



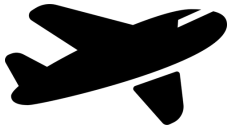

**Identify points different enough from majority, or from a reference**

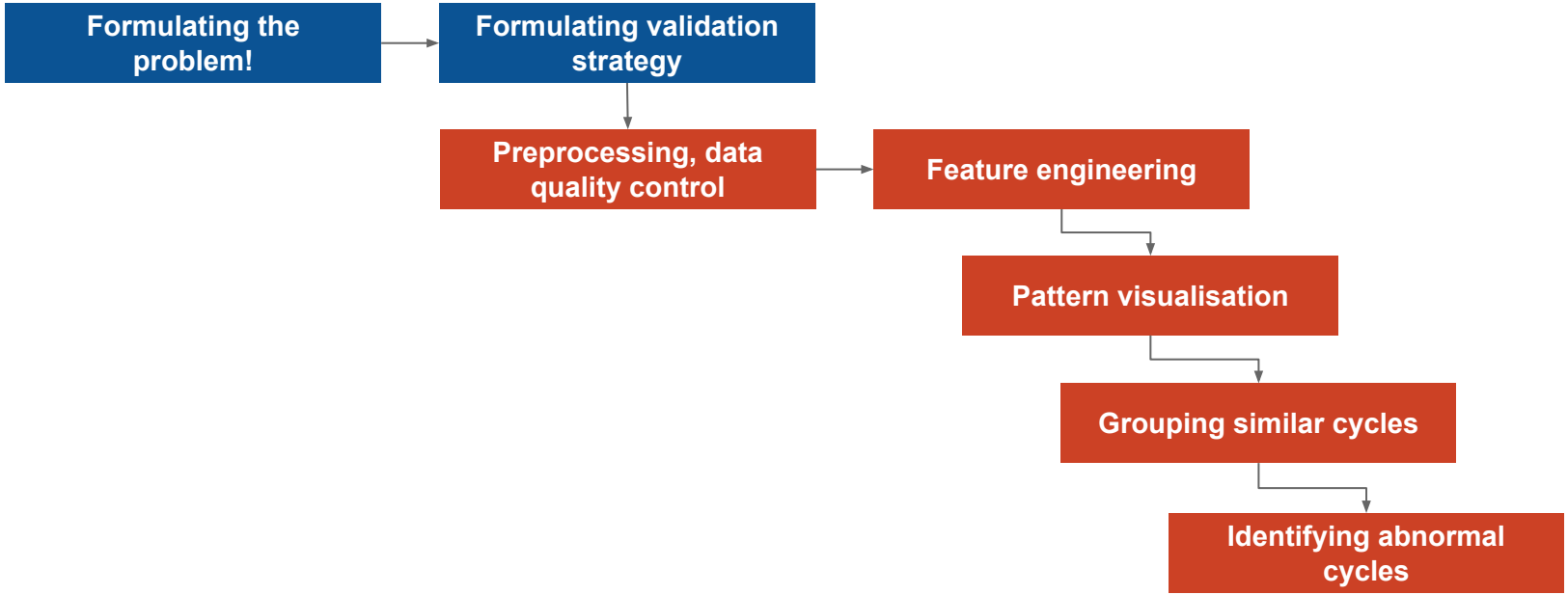
**Outlier detection, novelty detection, evaluate anomaly decision criteria**

**Isolation Forest**
**Neural networks to learn reference**

Focus here

# Time series use case follow-up



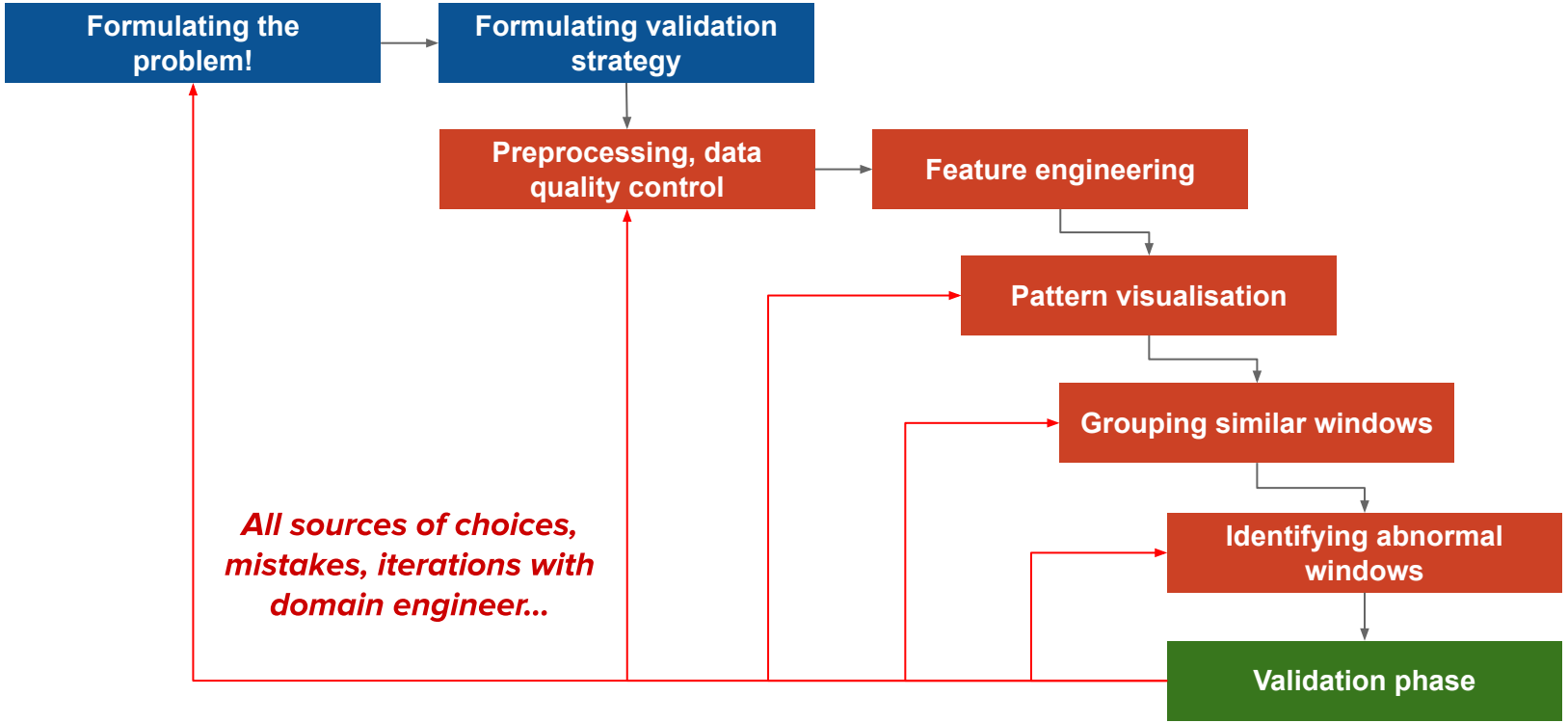


# Your turn!

It’s time to apply all of that together
with a bit of Python...

# Synthesis

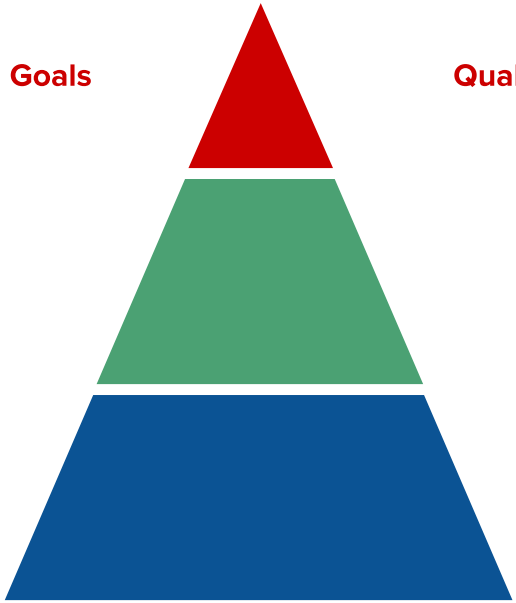
# Time series use case follow-up



# Quick piece of advice, again...

Machine Learning = complex field → a lot of: models, ideas, approaches, theories... every day!

How not to get lost? → Approach problems from global to detail




Qualitative → Quantitative

Global → Detail

Example:

**Identify flights different from the usual behavior**

**Unsupervised learning with dimensionality reduction and anomaly detection**

**Principal Components Analysis, Isolation Forest**

Many techniques can be used for the same means, including the famous: Deep Learning / Neural Networks, random forest, xgboost...

As AI decision-maker, you should: define clearly achievable goals, understand the means and their limitations

Questions?