

# Cours 2 - Introduction à l'Intelligence Artificielle

Alice Cohen-Hadria

Maîtresse de conférences à Sorbonne Université  
[alice.cohenhadria@gmail.com](mailto:alice.cohenhadria@gmail.com)

# Organisation du cours - Plan

## 1. **Données, visualisation**

- a. Indicateurs statistiques
- b. Visualisation
- c. Implémentation en Python

## 2. **Réduction de dimension**

- a. Matrice de covariance
- b. Analyse en Composantes Principales

## 3. **Classification**

- a. K plus proches voisins
- b. Réseaux de neurones

# Calcul matriciel



# Plan

1. Multiplication matricielle
2. Opérations matricielles
3. Projection (changement de base)
4. Vecteurs propres
  - a. Polynôme caractéristique
  - b. Valeur propre
  - c. Vecteur propre

# Multiplication matrice réel

Multiplication par un scalaire :

$$A = \begin{pmatrix} a_{11} & \dots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mm} \end{pmatrix}$$

$$\lambda A = \begin{pmatrix} \lambda a_{11} & \dots & \lambda a_{1m} \\ \vdots & \ddots & \vdots \\ \lambda a_{m1} & \dots & \lambda a_{mm} \end{pmatrix}$$

```
def mult_lambda (A, r):  
    nb_ligne, nb_col = A.shape  
    A_res = np.zeros(A.shape)  
  
    for i in range(nb_ligne):  
        for j in range(nb_col):  
            A_res[i][j] = r * A[i][j]  
    return A_res
```

# Produit matriciel 2

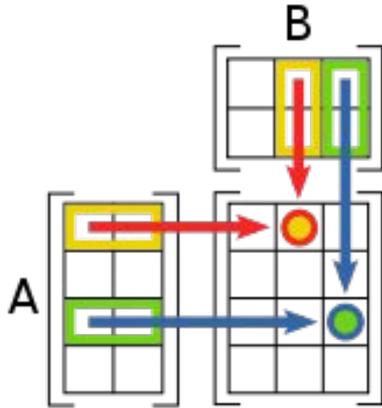
Multiplication de deux matrices A et B.

Soit  $A = (a_{ij})$  une matrice de dimension  $(m,n)$  et  $B = (b_{ij})$  une matrice de dimension  $(n,p)$ .

Le produit C de dimension  $(m,p)$  de A et B est défini comme :

$$\forall i, j : c_{ij} = \sum_{k=1}^n a_{ik} b_{kj} = a_{i1} b_{1j} + a_{i2} b_{2j} + \cdots + a_{in} b_{nj}$$

# Produit matriciel



```
def dot(A,B):  
    m,n = A.shape  
    _,p = B.shape  
    res = np.zeros((m,p))  
  
    for i in range(m):  
        for j in range(p):  
            for k in range(n):  
                res[i][j] = res[i][j] + A[i][k]*B[k][j]  
    return res
```

# Norme euclidienne

Norme euclidienne d'un vecteur (c'est la longueur du vecteur). Elle se calcule comme :

$$\text{Soit } \mathbf{v} \in \mathbb{R}^n, \mathbf{v} = v_1, v_2, \dots, v_n$$

La norme euclidienne de  $\mathbf{v}$  est :

$$\|\mathbf{v}\| = \sqrt{\sum_{i=1}^n v_i^2}$$

# Norme euclidienne

Normalisation d'un vecteur :

On divise chacune de ces composantes par la norme du vecteur.

La norme du vecteur normalisé est de 1.

```
def norme(v):  
    n, = v.shape  
    res = 0  
    for i in range(n):  
        res = res + v[i]**2  
    return np.sqrt(res)
```

# Déterminant

Utile pour l'inversion matricielle  $A^{-1}$

Une matrice est inversible si son déterminant est non nul.

On s'en servira pour calculer le polynôme caractéristique.

# Determinant

Déterminant d'une matrice de dimension 2

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc.$$

Déterminant d'une matrice de dimension 3

$$\begin{aligned} |A| &= \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} = a \begin{vmatrix} \square & \square & \square \\ \square & e & f \\ \square & h & i \end{vmatrix} - b \begin{vmatrix} \square & \square & \square \\ d & \square & f \\ g & \square & i \end{vmatrix} + c \begin{vmatrix} \square & \square & \square \\ d & e & \square \\ g & h & \square \end{vmatrix} = a \begin{vmatrix} e & f \\ h & i \end{vmatrix} - b \begin{vmatrix} d & f \\ g & i \end{vmatrix} + c \begin{vmatrix} d & e \\ g & h \end{vmatrix} \\ &= aei + bfg + cdh - ceg - bdi - afh. \end{aligned}$$

# Propriété du déterminant

Le déterminant d'une matrice est nul si et seulement si cette matrice est non inversible.

# Covariance

Extension de la notion de variance.

$$V = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2.$$

Soit deux vecteurs  $X = \{x_1, x_2, \dots, x_n\}$  et  $Y = \{y_1, y_2, \dots, y_n\}$ . Leur covariance s'écrit :

$$\text{cov}(X, Y) = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{X})(y_i - \bar{Y}).$$

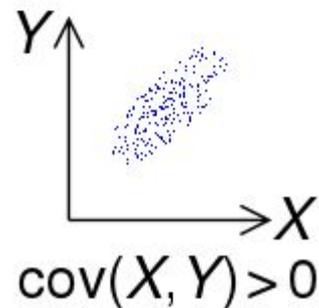
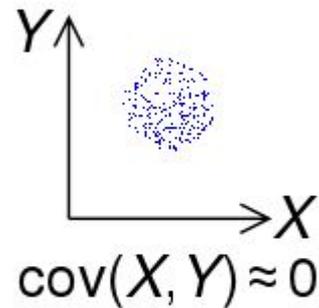
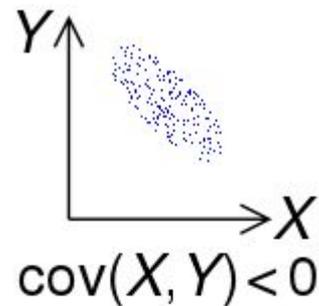
# Covariance

Permet de quantifier leurs écarts conjoints par rapport à leurs espérances respectives.

Le signe de la covariance montre donc la tendance de la relation linéaire entre X et Y.

Si les vecteurs sont normalisés, alors, plus la covariance est élevée, plus la relation entre X et Y est forte.

La covariance d'un vecteur avec lui même est la variance du vecteur.



# Covariance

```
def covariance(x, y):  
    n, = x.shape  
    moy_x = np.mean(x)  
    moy_y = np.mean(y)  
  
    res = 0  
  
    for i in range(n):  
        res = res + (x[i] - moy_x)*(y[i] - moy_y)  
    return 1/(n-1) * res
```

# Covariance - Probabilité.

On peut voir un vecteur d'une manière probabiliste.

Un vecteur est alors composé d'un certain nombre de tirage (ou observation) de cette variable aléatoire. La moyenne du vecteur est alors appelée espérance.

Dans ce cas, on dit que quand la covariance entre deux variables aléatoires est nulle, alors les deux variables sont indépendantes.

# Matrice de covariance

La matrice de covariance d'un vecteur de  $p$  composantes  $\vec{X} = \begin{pmatrix} X_1 \\ \vdots \\ X_p \end{pmatrix}$

dont chacune possède une variance, est la matrice carrée dont le terme générique est donné par:

$$a_{i,j} = \text{Cov}(X_i, X_j)$$

## Matrice de covariance

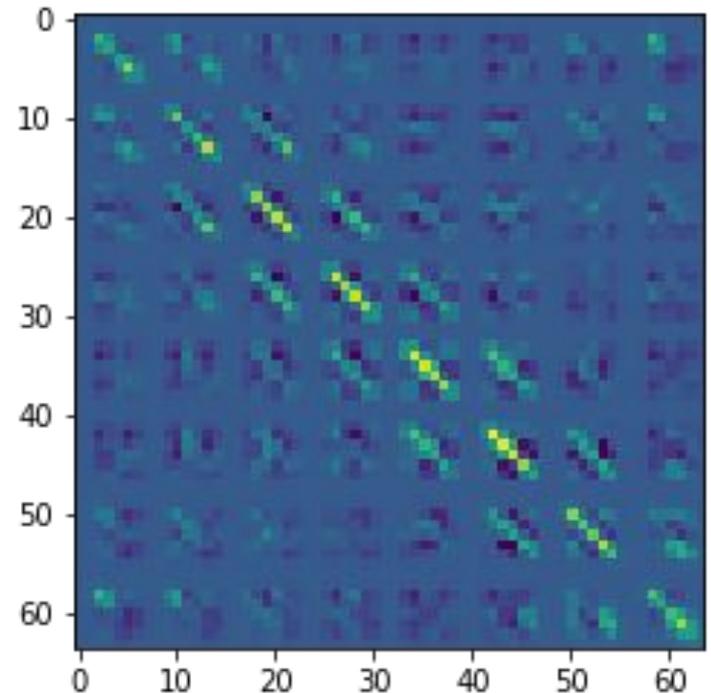
$$\text{Var}(\vec{X}) = \begin{pmatrix} \text{Var}(X_1) & \text{Cov}(X_1, X_2) & \cdots & \text{Cov}(X_1, X_p) \\ \text{Cov}(X_2, X_1) & \ddots & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \text{Cov}(X_p, X_1) & \cdots & \cdots & \text{Var}(X_p) \end{pmatrix}$$

# Matrice de covariance

Définie positive

Symétrique

Sur la diagonale, on retrouve les variances.



# Matrice diagonale

Une matrice carrée  $D = (d_{i,j})_{1 \leq i,j \leq n}$  est dite diagonale si :

$$\forall (i, j) \in \llbracket 1, n \rrbracket^2, i \neq j \Rightarrow d_{i,j} = 0.$$

$$\text{diag}(a_1, a_2, \dots, a_n) = \begin{pmatrix} a_1 & 0 & \dots & 0 \\ 0 & a_2 & \dots & \vdots \\ \vdots & \dots & \dots & 0 \\ 0 & \dots & 0 & a_n \end{pmatrix}.$$

# Diagonalisation de matrice

Quand on parle de diagonaliser une matrice carrée  $A$ , on cherche en fait deux matrices  $P$  et  $D$  telle que :

$$A = PDP^{-1}$$

On cherche une nouvelle base dans laquelle  $A$  est diagonale.  
La matrice  $P$  est dite de changement de base.

# Diagonalisation d'une matrice

Plusieurs étapes :

1. Calcul du polynôme caractéristique
2. Calcul des valeurs propres
3. Calcul des vecteurs propres
4. Diagonalisation

# Vecteur propres

Def :

Un vecteur  $x$  est un vecteur propre de la matrice carrée  $A$  de taille  $n \times n$  si :

$$Ax = \lambda x$$

Le réel  $\lambda$  est une valeur propre de  $A$  (si il y a une solution non triviale, autre que zéro à l'équation ci-dessus).

On dit d'un vecteur propre qu'il est associé à la valeur propre  $\lambda$ .

# Propriétés des vecteur propres

Si  $v_1, v_2, \dots, v_r$  sont des vecteurs propres associés à des valeurs propres distinctes  $\lambda_1, \lambda_2, \dots, \lambda_r$  d'une matrice  $A$  alors  $\{v_1, v_2, \dots, v_r\}$  sont linéairement indépendants.

On ne peut exprimer aucun des vecteurs propres avec une combinaison linéaire des autres.

# Comment calculer les vecteurs propres ?

On a :

$$Ax = \lambda x$$

$$(A - \lambda I)x = 0$$

$x$  doit être non nul, donc  $(A - \lambda I)x = 0$  doit avoir des solutions non triviales.

Donc  $(A - \lambda I)$  n'est pas inversible, et :

$$\det(A - \lambda I) = 0$$

On doit résoudre cette équation pour trouver les valeurs propres.

# Définitions

Pour une matrice carrée  $A$ , on définit :

- Son polynôme caractéristique par  $\det(A - \lambda I)$
- Son équation caractéristique par  $\det(A - \lambda I) = 0$

# Valeurs propres

Les valeurs propres sont les racines du polynôme caractéristique.

- Si les valeurs propres sont complexes, alors  $A$  n'est pas diagonalisable sur  $\mathbb{R}$
- Si on a une seule racine,  $A$  est diagonalisable si et seulement si  $A$  est diagonale au départ.
- Si on a deux racines, alors  $A$  est diagonalisable et sa forme diagonale est :

$$\Delta = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$$

## Exemple

$$A = \begin{pmatrix} 6 & -1 \\ 2 & 3 \end{pmatrix}$$

$$\lambda_1 = 4 \quad \lambda_2 = 5$$

# Trouver les vecteurs propres

On repart de l'équation:

$$(A - \lambda I)x = 0$$

On remplace les lambda par les valeurs calculées avec le polynôme caractéristique.

On résout l'équation et on trouve les vecteurs propres. Il y a toujours une infinité de solutions, on choisit des valeurs "qui marchent" (souvent les plus simples possibles).

## Exemple

$$A = \begin{pmatrix} 6 & -1 \\ 2 & 3 \end{pmatrix}$$

$$x_1 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

$$x_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

# Diagonalisation de matrice

En general,

$$A(\mathbf{v}_1 \quad \mathbf{v}_2 \quad \cdots \quad \mathbf{v}_n) = (\mathbf{v}_1 \quad \mathbf{v}_2 \quad \cdots \quad \mathbf{v}_n) \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \\ 0 & 0 & \cdots & \lambda_n \end{pmatrix}$$

et si  $(\mathbf{v}_1 \quad \mathbf{v}_2 \quad \cdots \quad \mathbf{v}_n)$  est inversible,  $A$  vaut

$$(\mathbf{v}_1 \quad \mathbf{v}_2 \quad \cdots \quad \mathbf{v}_n) \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \\ 0 & 0 & \cdots & \lambda_n \end{pmatrix} (\mathbf{v}_1 \quad \mathbf{v}_2 \quad \cdots \quad \mathbf{v}_n)^{-1}$$

# Diagonalisation de matrice

On veut exprimer  $A$  comme:

$$A = PDP^{-1}$$

$P$  est donc la matrice qui a les vecteurs propres en colonnes.

$D$  est la matrice avec les valeurs propres en diagonales.

# Diagonalisation de matrice

## Théorème de diagonalisation

Une matrice  $A$  de taille  $n \times n$  est diagonalisable si et seulement si  $A$  n'a pas de vecteurs propres linéairement dépendants.

Rappel: Les vecteurs propres sont linéairement indépendants.

## Exemple

$$A = \begin{pmatrix} 6 & -1 \\ 2 & 3 \end{pmatrix} \quad P = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix} \quad D = \begin{pmatrix} 5 & 0 \\ 0 & 4 \end{pmatrix}$$

# Changement de base

En multipliant  $A$  par la matrice de passage  $P$ , on **projette la matrice  $A$**  dans une nouvelle base où elle est diagonale (matrice  $D$ ).

On peut donc **changer la représentation de données** (matrice  $A$ ), en utilisant la matrice de passage  $P$ , pour avoir de **nouvelles manières de représenter** nos données.

Les propriétés des valeurs propres et vecteurs propres peuvent être intéressantes à exploiter. C'est ce qui est fait dans **l'analyse en composantes principales**.

ACP

# Données en ACP

En ACP les données sont dans une matrice  $X$  à  $n$  lignes et  $p$  colonnes où:

- Chaque ligne représente un individu
- Chaque colonne représente une variable

Les variables sont quantitatives : la matrice  $A$  est constituée de valeurs numériques.

# Données en ACP

$X$  est une matrice  $n \times p$  de valeurs numériques :

$$X = \begin{bmatrix} x_{11} & \cdot & \cdot & \cdot & \cdot & \cdot & x_{1p} \\ x_{21} & \cdot & \cdot & \cdot & \cdot & \cdot & x_{2p} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ x_{n1} & \cdot & \cdot & \cdot & \cdot & \cdot & x_{np} \end{bmatrix}$$

# Données en ACP

Un individu est un élément de  $\mathbb{R}^p$

Le  $j^{\text{ème}}$  individu :

$$X = \begin{bmatrix} x_{11} & \cdot & \cdot & \cdot & \cdot & \cdot & x_{1p} \\ x_{21} & \cdot & \cdot & \cdot & \cdot & \cdot & x_{2p} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ x_{i1} & \cdot & \cdot & \cdot & x_{ij} & \cdot & x_{ip} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ x_{n1} & \cdot & \cdot & \cdot & \cdot & \cdot & x_{np} \end{bmatrix}$$

# Données en ACP

Une variable est un élément de  $\mathbb{R}^n$

La  $j^{\text{ème}}$  variable :

$$X = \begin{bmatrix} x_{11} & \cdot & \cdot & \cdot & x_{1j} & \cdot & x_{1p} \\ x_{21} & \cdot & \cdot & \cdot & \cdot & \cdot & x_{2p} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & x_{ij} & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ x_{n1} & \cdot & \cdot & \cdot & x_{nj} & \cdot & x_{np} \end{bmatrix}$$

# Analyse en composante principale

ACP (ou PCA en anglais).

On suppose qu'on a une collection de  $m$  points  $\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$

Chacun de ces points est un vecteur dans  $\mathbb{R}^n$

On veut appliquer un algorithme de compression, ou réduction de dimension, avec perte.

-> Stockage des points avec moins de mémoire, et perte de précision

On veut perdre le moins de précision possible.

# ACP - Principe

On veut encoder chacun de ces points dans une dimension inférieure.

Pour chaque point  $x^i \in \mathbb{R}^n$  on veut trouver un vecteur “code” correspondant  $c^i \in \mathbb{R}^l$

Si  $l$  est plus petit que  $n$ , alors stocker les vecteurs code sera moins coûteux.

# ACP - première étape

Il faut centrer la matrice de données  $X$  de telles sortes à ce que chaque variable (donc chaque colonne) ait une moyenne nulle.

On commence par calculer la moyenne par colonne :

$$\begin{array}{ccccccc} \left[ \begin{array}{ccccccc} x_{11} & \cdot & \cdot & \cdot & x_{1j} & \cdot & x_{1p} \\ x_{21} & \cdot & \cdot & \cdot & \cdot & \cdot & x_{2p} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & x_{ij} & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ x_{n1} & \cdot & \cdot & \cdot & x_{nj} & \cdot & x_{np} \end{array} \right] \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ \bar{x}_1 & \cdot & \cdot & \cdot & \bar{x}_j & \cdot & \bar{x}_p \end{array}$$

# ACP - première étape

Et ensuite on peut centrer notre matrice :

$$X = \begin{bmatrix} x_{11} - \bar{x}_1 & \cdot & \cdot & x_{1j} - \bar{x}_j & \cdot & x_{1p} - \bar{x}_p \\ x_{21} - \bar{x}_1 & \cdot & \cdot & \cdot & \cdot & x_{2p} - \bar{x}_p \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & x_{ij} - \bar{x}_j & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ x_{n1} - \bar{x}_1 & \cdot & \cdot & x_{nj} - \bar{x}_j & \cdot & x_{np} - \bar{x}_p \end{bmatrix}$$

# ACP - première étape

Il faut aussi réduire la matrice de telle sorte à ce que chaque variable ait un écart type de 1.

On commence par calculer les écart-types pour chaque variable (par colonne)

$$\begin{array}{ccccccc} \left[ \begin{array}{ccccccc} x_{11} & \cdot & \cdot & \cdot & x_{1j} & \cdot & x_{1p} \\ x_{21} & \cdot & \cdot & \cdot & \cdot & \cdot & x_{2p} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & x_{ij} & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ x_{n1} & \cdot & \cdot & \cdot & x_{nj} & \cdot & x_{np} \end{array} \right] \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ \sigma_1 & \cdot & \cdot & \cdot & \sigma_j & \cdot & \sigma_p \end{array}$$

# ACP - première étape

Après centrage on peut ensuite réduire la matrice X :

$$X = \begin{bmatrix} (x_{11} - \bar{x}_1)/\sigma_1 & \cdot & \cdot & (x_{1j} - \bar{x}_j)/\sigma_j & \cdot & (x_{1p} - \bar{x}_p)/\sigma_p \\ (x_{21} - \bar{x}_1)/\sigma_1 & \cdot & \cdot & \cdot & \cdot & (x_{2p} - \bar{x}_p)/\sigma_p \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & (x_{ij} - \bar{x}_j)/\sigma_j & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ (x_{n1} - \bar{x}_1)/\sigma_1 & \cdot & \cdot & (x_{nj} - \bar{x}_j)/\sigma_j & \cdot & (x_{np} - \bar{x}_p)/\sigma_p \end{bmatrix}$$

On dit que la matrice X est centrée et réduite.

# Objectifs

On considère  $X$  centrée et réduite.

On peut analyser  $X$  à travers ses lignes (individus), ou ses colonnes (variables).

-> On veut résumer l'information contenue dans  $X$

# Grouper les individus et les variables

Typologie des individus :

- Certains individus varie sur certaines variable
- On veut former des groupes d'individus semblables
- Terme: ressemblance

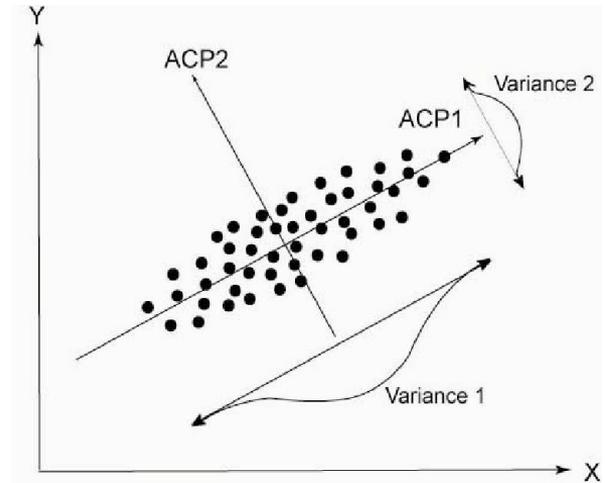
Typologie des variables:

- Il existe des variables liées entre elles
- On veut former des groupes de variables liées
- Terme : liaison, corrélation

# Représentation

On cherche à trouver les deux axes ACP1 et ACP2.

On projettera ensuite nos données dans cette nouvelles base.



# Variance = information

Personne	Taille
Alex	145
Ben	160
Chris	180



A



B



C

# Variance = information

Personne	Taille
Daniel	172
Elsa	173
Fernandez	171



A



B



C

# Variance = information

Ajoutons le poids des personnes:

Personne	Taille	Poids
Alex	145	68
Ben	160	67
Chris	180	69

On a maintenant le “double” d’information (taille + poids).

Pourtant, pour différencier ces trois personnes, on va majoritairement se servir de la taille. On a “réduit” nos données de 2D à 1D.

C’est l’intuition derrière l’ACP.

# Matrice de covariance

En calculant la matrice de covariance sur nos données, par colonne, on a une indication de la variance ENTRE nos variance (la covariance).

La matrice de covariance est carrée, symétrique et réelle. Elle est donc diagonalisable.

En diagonalisant la matrice de covariance, on réduit les corrélations entre les données, dans le nouvel espace.

La matrice de covariance par colonne a pour dimension  $(p \times p)$

# Diagonalisation de la matrice de covariance

On cherche les valeurs propres et vecteurs propres de la matrice de covariance.

Les valeurs propres sont les variances dans la nouvelle base.

On choisit les vecteurs propres qui possèdent les plus hautes valeurs propres.

En général, on peut choisir soit un nombre  $k$  de vecteurs propres a priori (si on veut des données dans une certaines dimensions), soit une variance totale de 80 (ou autre) %.

# Variance expliquée

Chaque valeur propre de notre décomposition représente la variance de cette composante de cette représentation.

On parle alors de variance expliquée.

Soit  $N$  vecteurs propres de la matrice de covariance, associés à  $N$  valeurs propres. La variance expliquée d'une valeur propre  $\lambda_i$  s'exprime comme :

$$\frac{\lambda_i}{\lambda_1 + \lambda_2 + \dots + \lambda_n}$$

# Projection des données

Une fois les vecteurs propres de la matrice de covariance choisis, on projette notre matrice de données  $X$  dans ce nouvel espace.

Appelons  $P$  la matrice formée des vecteurs propres (en colonnes). Elle est de taille  $p$  par  $k$  (le nombre de vecteurs propres choisis).

La projection s'obtient en multipliant  $X$  par  $P$ . On obtient une nouvelle matrice de taille  $n$  par  $k$ .

# Projection des données

$$D = (v_1, v_2, \dots, v_k)$$

$$X = \begin{bmatrix} x_{11} & \cdot & \cdot & \cdot & x_{1j} & \cdot & x_{1p} \\ x_{21} & \cdot & \cdot & \cdot & \cdot & \cdot & x_{2p} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & x_{ij} & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ x_{n1} & \cdot & \cdot & \cdot & x_{nj} & \cdot & x_{np} \end{bmatrix}$$

$$\tilde{X} = \begin{pmatrix} \tilde{x}_{11} & \cdot & \cdot & \cdot & \tilde{x}_{1k} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \tilde{x}_{n1} & \cdot & \cdot & \cdot & \tilde{x}_{nk} \end{pmatrix}$$

# Nouvelle représentation

Si  $k < p$ , alors on a réduit la taille de nos données.

Mais on les a réduit de telle sorte à exprimer un MAXIMUM de variance dans notre nouvelle base (notre nouvelle représentation).

# Algorithme ACP

1. Centrer et réduire nos données
2. Calculer la matrice de covariance
3. On cherche à diagonaliser cette matrice pour réduire les corrélations entre variables
  - Recherche des valeurs propres et des vecteurs propres
  - Sélection des vecteurs propres avec les plus grandes valeurs propres (donc les plus grandes variances dans la nouvelles base)
4. Projeter nos données dans cette nouvelle base

# ACP - Implémentation

Avec numpy c'est très simple à implémenter:

- Centrée réduite :
- Matrice de covariance : `np.cov`
  - Attention à bien faire la matrice par colonne
- Valeurs propres vecteurs propres : `np.eigh`
  - On peut spécifier les valeurs propres qu'on veut (directement avoir les plus grandes)
- Classer les valeurs propres : `np.argsort`
- Projection : `np.matmul`

# ACP - Visualisation

