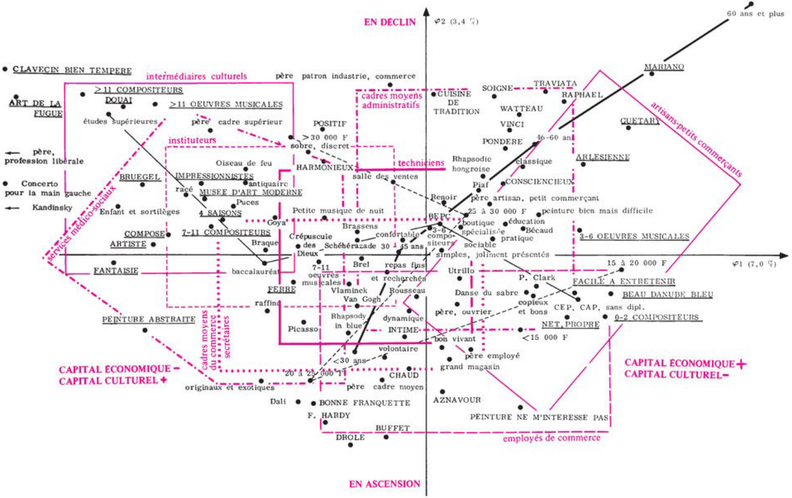
Introduction to data science & artificial intelligence (IF7100)

Arthur Charpentier

#271 Multivariate Analysis: Projections

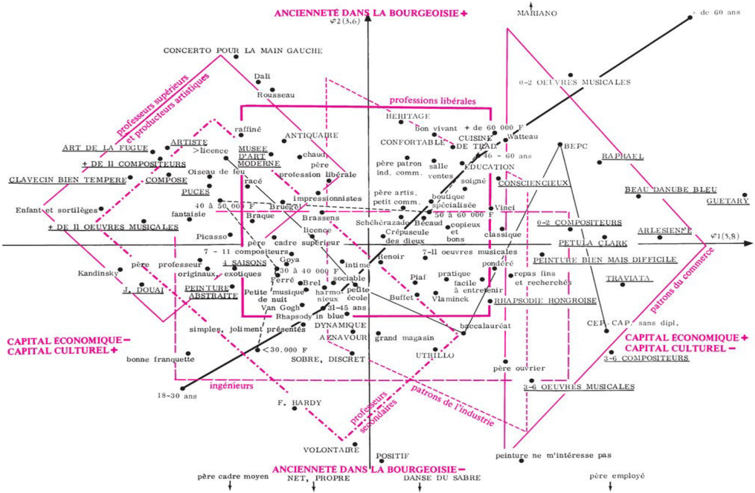
été 2020

Projections



in La Distinction (critique sociale du jugement), Pierre Bourdieu

Projections



in La Distinction (critique sociale du jugement), Pierre Bourdieu

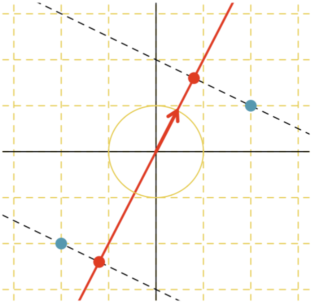
(Orthogonal) Projection

Let $\boldsymbol{x} \in \mathbb{R}^d$ and $\vec{\boldsymbol{u}} \in \mathbb{R}^d$ with $\|\vec{\boldsymbol{u}}\| = 1$.
Projection on $\vec{\boldsymbol{u}}$ of $\vec{\boldsymbol{u}}$ is $\langle \vec{\boldsymbol{u}}, \boldsymbol{x} \rangle\ \vec{\boldsymbol{u}}$

If we map our data on one dimension ($\vec{\boldsymbol{u}}$)
point $\boldsymbol{x}$ is now $\boldsymbol{x}' = \boldsymbol{u}^\top \boldsymbol{x} = \langle \vec{\boldsymbol{u}}, \boldsymbol{x} \rangle$

Variance of $\boldsymbol{x}'$s is $\boldsymbol{u}^\top \text{Var}(\boldsymbol{X})\boldsymbol{u}$
In which direction $\vec{\boldsymbol{u}}$ is the variance maximal ?

Maximal when $\vec{\boldsymbol{u}}$ is the eigenvector of $\text{Var}(\boldsymbol{X})$
associated with the largest eigenvector.
Called principal component



(Orthogonal) Projection

If we want to map data $\boldsymbol{X}$ from dimension d to (just) dimension k, to capture as much variance as possible,

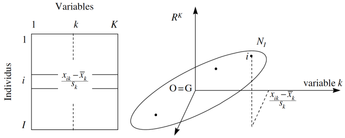
$$\boldsymbol{x} \mapsto \left(\boldsymbol{u}_1^\top \boldsymbol{x}, \cdots, \boldsymbol{u}_k^\top \boldsymbol{x}\right) = \begin{pmatrix} -\boldsymbol{u}_1^\top- \\ -\boldsymbol{u}_2^\top- \\ \vdots \\ -\boldsymbol{u}_k^\top- \end{pmatrix} \begin{pmatrix} | \\ \boldsymbol{x} \\ | \end{pmatrix}$$

$\vec{\boldsymbol{u}}_1, \cdots, \vec{\boldsymbol{u}}_k$ are eigenvalues, $\lambda_1 \geq \cdots \geq \lambda_k$ of $\text{Var}(\boldsymbol{X}) = \dfrac{1}{n}\boldsymbol{X}^\top \boldsymbol{X}$

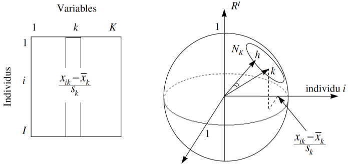
Principal Component Analysis

n individuals, k variables

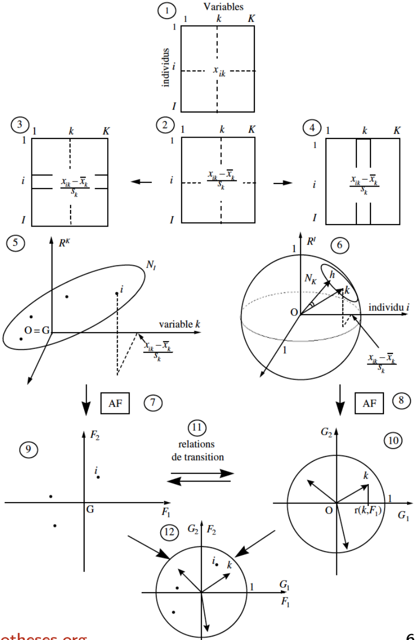
$\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n \in \mathbb{R}^k$



$\boldsymbol{x}_1, \cdots, \boldsymbol{x}_k \in \mathbb{R}^n$



source: Analyses factorielles simples et multiples



Decathlon

```
> library(FactoMineR)
> data(decathlon)
> head(decathlon[,1:10])
          100m Long.jump Shot.put H.jump  400m 110m.hd
SEBRLE   11.04      7.58    14.83   2.07 49.81   14.69
CLAY     10.76      7.40    14.26   1.86 49.37   14.05
KARPOV   11.02      7.30    14.77   2.04 48.37   14.09
BERNARD  11.02      7.23    14.25   1.92 48.93   14.99
YURKOV   11.34      7.09    15.19   2.10 50.42   15.31
WARNERS  11.11      7.60    14.31   1.98 48.68   14.23
```

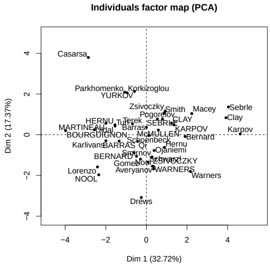
Decathlon

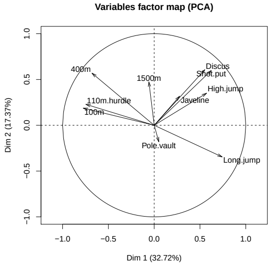
In matrix $\boldsymbol{X}$,

- rows are individuals
- columns are variables

Consider projections of individuals (points in $\mathbb{R}^{10}$) and variables (points in $\mathbb{R}^n$) on the first two (princiapl) components.

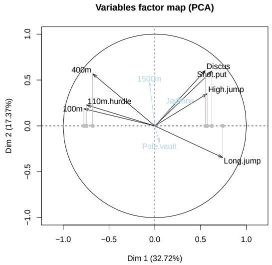
```
> pca <- PCA(decathlon[,1:10])
> plot(pca,choix="ind")
> plot(pca,choix="var")
```





Decathlon

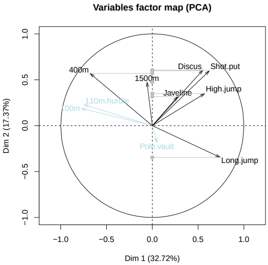
```
> dimdesc(pca)
$Dim.1
$Dim.1$quanti
            correlation      p.value
Long.jump     0.7418997 2.849886e-08
Shot.put      0.6225026 1.388321e-05
High.jump     0.5719453 9.362285e-05
Discus        0.5524665 1.802220e-04
400m         -0.6796099 1.028175e-06
110m.hurdle  -0.7462453 2.136962e-08
100m         -0.7747198 2.778467e-09
```



Because variables were normalized, projections of variables always belong to unit disk.

Decathlon

```
> dimdesc(pca)
$Dim.2
$Dim.2$quanti
             correlation      p.value
Discus         0.6063134 2.650745e-05
Shot.put       0.5983033 3.603567e-05
400m           0.5694378 1.020941e-04
1500m          0.4742238 1.734405e-03
High.jump      0.3502936 2.475025e-02
Javeline       0.3169891 4.344974e-02
Long.jump     -0.3454213 2.696969e-02
```



Decathlon

```
> pca$eig
         eigenvalue    percentage cumulative percentage
                      of variance             of variance
comp 1        3.272        32.719                  32.719
comp 2        1.737        17.371                  50.090
comp 3        1.405        14.049                  64.140
comp 4        1.057        10.569                  74.708
comp 5        0.685         6.848                  81.556
```