Cours 5 : Analyse en Composantes Principales

1 Le problème mathématique et sa solution

On dispose de n observations de p variables *continues*, qui sont les éléments de la matrice suivante :

$$X = \begin{pmatrix} x_{1,1} & \cdots & x_{1,p} \\ \vdots & \vdots & \vdots \\ x_{n,1} & \cdots & x_{n,p} \end{pmatrix}$$

On appelle

- les colonnes $x_1, \ldots, x_p \in \mathbb{R}^n$: ce sont les **variables**
- les lignes $a_1, \ldots, a_n \in \mathbb{R}^p$: ce sont les **observations**.

On voit $a_1, \ldots, a_n$ comme des points dans $\mathbb{R}^p$. On s'intéresse au problème suivant : Pour une dimension d donnée (en pratique, $d = 2$ par exemple), quel est le sous-espace affine de dimension d de $\mathbb{R}^p$ qui minimise la somme des carrés des distances aux observations $a_1, \ldots, a_n$?

Ce sous-espace de petite dimension permettra une représentation graphique des données (si $d = 2$) aussi fidèle que possible au nuage p-dimensionnel initial, afin de mieux les comprendre visuellement : identification d'observations proches entre elles (sous-groupes), distantes, ou de points extrêmes. La compréhension des directions du sous-espace affine donnera également une idée des principaux critères qui décrivent la distribution des données.

Cette question rappelle la régression linéaire (pour une composante $y = x_i$ donnée, on recherchait un hyperplan $\mathcal{H}$ minimisant la somme des carrés des distances entre les observations $a_1, \ldots, a_n$ et leurs projections sur $\mathcal{H}$ parallèlement à $\vec{e_i}$; on se demandait ensuite si on pouvait supposer que l'hyperplan était parallèle à certains axes, ce qui signifiait l'absence de dépendance de y vis-à-vis de ces variables), cependant ici on ne distingue pas une variable par rapport aux autres et on recherche un sous-espace de plus petite dimension ; plutôt qu'à l'*équation* du sous-espace, on s'intéressera à ses *vecteurs directeurs*.

1.1 Premières observations

On rappelle qu'un sous-espace affine $\mathcal{V}$ de $\mathbb{R}^p$ est un sous-espace vectoriel de $\mathbb{R}^p$ translaté par un élément de $\mathbb{R}^p$: $\mathcal{V} = v + V$ où $v \in \mathbb{R}^p$ est un élément quelconque de $\mathcal{V}$, et V est un sous-espace vectoriel de $\mathbb{R}^p$ (la "direction" de $\mathcal{V}$). On commence par se ramener à la recherche de V seul.

Notons $\overline{a} = \begin{pmatrix} \overline{x}_1 & \cdots & \overline{x}_p \end{pmatrix}$ le barycentre de $a_1, \ldots, a_n$.

Lemme : Si $\mathcal{V}$ minimise $\sum_{i=1}^n d(x_i; \mathcal{V})^2$ parmi les sous-espaces affines de dimension d, alors $\overline{a} \in \mathcal{V}$.

En effet, soit $\mathcal{V}$ un sous-espace affine de dimension d, et notons $\mathcal{V}'$ le sous-espace affine parallèle à $\mathcal{V}$ et passant par $\overline{a}$, autrement dit $\mathcal{V}' = \overline{a} + V$. Pour tout point $b \in \mathbb{R}^p$, les projections orthogonales de b sur $\mathcal{V}$ et $\mathcal{V}'$ sont reliées par

$$\mathrm{Proj}_{\mathcal{V}'}(b) = \mathrm{Proj}_{\mathcal{V}}(b) + \overline{a} - \mathrm{Proj}_{\mathcal{V}}(\overline{a}).$$

D'où :

$$\begin{aligned}\sum_{i=1}^n d(a_i;\mathcal{V}')^2 &= \sum_{i=1}^n |a_i - \mathrm{Proj}_{\mathcal{V}'}(a_i)|^2 \\ &= \sum_{i=1}^n |a_i - \mathrm{Proj}_{\mathcal{V}}(a_i)|^2 + \sum_{i=1}^n |\overline{a} - \mathrm{Proj}_{\mathcal{V}}(\overline{a})|^2 - 2\sum_{i=1}^n (a_i - \mathrm{Proj}_{\mathcal{V}}(a_i)) \cdot (\overline{a} - \mathrm{Proj}_{\mathcal{V}}(\overline{a})) \\ &= \sum_{i=1}^n d(a_i,\mathcal{V})^2 + n|\overline{a} - \mathrm{Proj}_{\mathcal{V}}(\overline{a})|^2 - 2(n\overline{a} - n\mathrm{Proj}_{\mathcal{V}}(\overline{a})) \cdot (\overline{a} - \mathrm{Proj}_{\mathcal{V}}(\overline{a})) \\ &= \sum_{i=1}^n d(a_i,\mathcal{V})^2 - n|\overline{a} - \mathrm{Proj}_{\mathcal{V}}(\overline{a})|^2 \leq \sum_{i=1}^n d(a_i,\mathcal{V})^2,\end{aligned}$$

avec égalité seulement si $\overline{a} \in \mathcal{V}$, ce qui montre que translater $\mathcal{V}$ pour que cet espace inclue $\overline{a}$ réduit strictement la somme des carrés des distances. Le minimiseur doit donc contenir $\overline{a}$.

On pourra donc supposer que $\overline{a} \in \mathcal{V}$ quand on cherchera le minimiseur : on a $\mathcal{V} = \overline{a} + V$ où l'espace vectoriel V des directions reste à déterminer. Remarquons maintenant que le minimum peut se traduire en un autre maximum :

Lemme : Parmi les sous-espaces affines de dimension d contenant $\overline{a}$, $\mathcal{V}$ minimise la somme des carrés des distances si, et seulement si $\mathcal{V}$ maximise l'**inertie** des projections orthogonales de $a_1, \ldots, a_n$ sur $\mathcal{V}$.

L'inertie d'observations $a_1, \ldots, a_n \in \mathbb{R}^p$ est

$$I(a_1, \ldots, a_n) = \sum_{i=1}^n |a_i - \overline{a}|^2.$$

En écrivant la norme (euclidienne) en fonction des composantes et en échangeant les sommes on constate que

$$I(a_1, \ldots, a_n) = \sum_{i=1}^n \sum_{j=1}^p (x_{ij} - \overline{x}_j)^2 = \sum_{j=1}^p \sum_{i=1}^n (x_{ij} - \overline{x}_j)^2 = n \sum_{j=1}^p \mathrm{Var}(x_j)$$

Ainsi, l'inertie est la somme des carrés des distances au barycentre, mais aussi la somme des variances de composantes dans une quelconque base orthonormée (ici, $x_1, \ldots, x_p$ sont les variables lues dans la base canonique $(e_1, \ldots, e_p)$ mais on pourrait écrire ci-dessus les composantes dans une autre base orthonormée). L'inertie apparaît donc comme un analogue en dimension p de la variance : c'est une mesure de la dispersion des points. Le lemme exprime ainsi que chercher à approcher au mieux les données par un sous-espace revient à maximiser la dispersion des projections sur ce sous-espace.

Notons $\widehat{a}_i = \mathrm{Proj}_{\mathcal{V}}(a_i)$.

Le lemme est une simple conséquence de l'orthogonalité $a_i - \widehat{a}_i \perp V$ donc $a_i - \widehat{a}_i \perp \widehat{a}_i - \overline{a}$: pour tout i,

$$d(a_i;\mathcal{V})^2 = |a_i - \widehat{a}_i|^2 = |a_i - \overline{a}|^2 - |\widehat{a}_i - \overline{a}|^2$$

d'où en sommant sur i :

$$\sum_{i=1}^n d(a_i;\mathcal{V})^2 = I(a_1, \ldots, a_n) - I(\widehat{a}_1, \ldots, \widehat{a}_n),$$

en utilisant le fait que le barycentre des projections $\widehat{a}_i$ est $\overline{a}$ (puisque la projection est affine, la moyenne des projections est la projection de la moyenne ; et la projection de $\overline{a}$ est a car $a \in \mathcal{V}$). Comme $I(a_1, \ldots, a_n)$ ne dépend pas de $\mathcal{V}$, minimiser $\sum_{i=1}^n d(a_i;\mathcal{V})^2$ revient à maximiser $I(\widehat{a}_1, \ldots, \widehat{a}_n)$.

Remarquons que l'inertie additionne les variances des variables, ce qui a peu de sens si les variables sont exprimées dans des unités différentes, et peut induire une sur-représentation injustifiée d'une variable. On choisira donc très souvent de **réduire** les données, c'est-à-dire de ramener leur variance à 1. Autrement dit, quitte à les centrer pour ramener $\overline{a}$ en l'origine, on les divise ensuite par leur écart-type. On parlera d'"ACP normée" dans ce cas. Ce choix peut être pertinent même si les variables sont exprimées dans la même unité, mais d'amplitudes variées.

1.2 Dimension zéro

Un sous-espace affine de dimension 0 est un singleton. Le minimiseur doit contenir $\overline{a}$ par le premier lemme, et en dimension 0 il n'y a qu'une direction possible, donc le minimiseur est $\{\overline{a}\}$.

Autrement dit, $\overline{a}$ est le point qui minimise la somme des carrés des distances aux observations. En ce sens, c'est le point approchant le mieux les observations.

1.3 Dimension un

On cherche la droite affine $\mathcal{V} = \overline{a} + \mathbb{R}\vec{u}$, avec $|\vec{u}| = 1$, qui maximise l'inertie des projections de $a_1, \dots, a_n$. Ici, la projection de a_i est $\widehat{a}_i = \overline{a} + (a_i - \overline{a}) \cdot \vec{u}\,\vec{u}$, d'où

$$I(\widehat{a}_1, \dots, \widehat{a}_n) = \sum_{i=1}^{n} |\widehat{a}_i - \overline{a}|^2 = \sum_{i=1}^{n} \big((a_i - \overline{a}) \cdot \vec{u}\big)^2 = \sum_{i=1}^{n} \vec{u}\,{}^t(a_i - \overline{a})(a_i - \overline{a})\,{}^t\vec{u} = n\vec{u}\Gamma\,{}^t\vec{u},$$

en voyant a_i et $\vec{u}$ comme des vecteurs-lignes (et en écrivant $\vec{u} \cdot \vec{v} = \vec{u}\,{}^t\vec{v} = \vec{v}\,{}^t\vec{u}$ pour $\vec{u}, \vec{v} \in \mathbb{R}^p$), et où

$$\Gamma = \frac{1}{n}\sum_{i=1}^{n} {}^t(a_i - \overline{a})(a_i - \overline{a}) = (\gamma_{jk})_{1 \le j,k \le p} \qquad \text{où} \qquad \gamma_{jk} = \frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \overline{x}_j)(x_{ik} - \overline{x}_k) = \mathrm{Cov}(x_j, x_k),$$

c'est-à-dire que Γ est la matrice de covariance des données.

Cette matrice Γ est symétrique, positive (au sens où $\vec{u}\Gamma\,{}^t\vec{u} \ge 0$ pour tout vecteur ligne $\vec{u} \in \mathbb{R}^p$), elle se diagonalise donc en base orthonormale : notons $\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_p \ge 0$ ses valeurs propres (en pratique, elles sont toutes distinctes), et $\vec{v}_1, \dots, \vec{v}_p \in \mathbb{R}^p$ une base orthonormale de vecteurs propres associée, vus comme des vecteurs lignes.

Si on décompose $\vec{u} = \sum_{j=1}^{p} u_j \vec{v}_j$, alors

$$\vec{u}\Gamma\,{}^t\vec{u} = \sum_{j=1}^{p}\sum_{k=1}^{p} u_j u_k \vec{v}_j \Gamma\,{}^t\vec{v}_k = \sum_{j=1}^{p} \lambda_j u_j^2,$$

en utilisant l'orthonormalité de $\vec{v}_1, \dots, \vec{v}_p$ et $\Gamma\,{}^t\vec{v}_k = \lambda_k\,{}^t\vec{v}_k$.

On est donc amené à déterminer les réels $u_1, \dots, u_p$, avec $\sum_{j=1}^{p} u_j^2 = 1$, qui maximise $\sum_{j=1}^{p} \lambda_j u_j^2$. Le maximum est λ_1, obtenu pour $u_1 = 1$ et $u_2 = \cdots = u_n = 0$ car $\sum_{j=1}^{p} \lambda_j u_j^2 \le \sum_{j=1}^{p} \lambda_1 u_j^2 = \lambda_1$.

Ainsi, la droite affine solution du problème est $\mathcal{V}_1 = \overline{a} + \mathbb{R}\vec{v}_1$, où $\vec{v}_1$ est un vecteur propre (unitaire) associé à la plus grande valeur propre de la matrice de covariance des données.

On appelle $\vec{v}_1$ la **première direction principale** du nuage de points $a_1, \dots, a_p$.

1.4 Dimensions supérieures

Plus généralement, le sous-espace affine de dimension d qui minimise la somme des carrés des distances aux observations est

$$\mathcal{V}_d = \overline{a} + \mathrm{Vect}(\vec{v}_1, \dots, \vec{v}_d),$$

où $\vec{v}_1, \dots, \vec{v}_d$ sont les vecteurs propres associées aux d plus grandes valeurs propres $\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_d$ de la matrice de covariance Γ. Ce sont les **directions principales** de $a_1, \dots, a_p$.

En effet, si $\mathcal{V} = \overline{a} + V$ est un sous-espace affine de $\mathbb{R}^p$ de dimension d, en notant $(\vec{u}_1, \dots, \vec{u}_d)$ une base orthonormée de V, et pour tout i en notant $\vec{u}_i = \sum_{j=1}^{p} u_{ij}\vec{v}_j$ la décomposition de $\vec{u}_i$ dans la base $(\vec{v}_1, \dots, \vec{v}_p)$ (c'est-à-dire $u_{ij} = \vec{u}_i \cdot \vec{v}_j$), alors la projection d'un point $a \in \mathbb{R}^p$ sur $\mathcal{V}$ est

$$\mathrm{Proj}_{\mathcal{V}}(a) = \overline{a} + \sum_{j=1}^{d} (a - \overline{a}) \cdot \vec{u}_j\ \vec{u}_j,$$

d'où l'inertie des projections sur $\mathcal{V}$, de la même façon que dans la section précédente, et en utilisant aussi l'orthonormalité des $\vec{u}_j$,

$$I_{\mathcal{V}} = \sum_{i=1}^{n}\sum_{j=1}^{d}((a_i-\overline{a})\cdot\vec{u}_j)^2 = n\sum_{j=1}^{d} {}^t\vec{u}_j\Gamma{}^t\vec{u}_j = n\sum_{j=1}^{d}\sum_{k=1}^{p}\lambda_k u_{jk}^2.$$

Cette somme se réécrit, en changeant l'ordre de sommation,

$$I_{\mathcal{V}} = n\sum_{k=1}^{p}\left(\lambda_k\sum_{j=1}^{d}u_{jk}^2\right).$$

D'une part, par orthonormalité des $(u_{jk})_{j,k}$, le coefficient devant λ_k est ≤ 1 ; d'autre part, la somme de tous ces coefficients vaut $\sum_{k=1}^{p}\sum_{j=1}^{d}u_{jk}^2 = \sum_{j=1}^{d}|\vec{u}_j|^2 = d$. Comme $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0$, on en déduit que $I_{\mathcal{V}}$ serait maximale si les d premiers coefficients prenaient leur valeur maximale 1 (et les autres étant donc égaux à 0), ce qui est réalisé par $\vec{u}_1 = \vec{v}_1$,..., $\vec{u}_d = \vec{v}_d$.

Pour une preuve formelle du fait que la somme est maximale pour des coefficients égaux à 1,...,1 (d fois) puis 0,...,0, on pourrait montrer que l'on peut trouver des coefficients qui augmentent la somme dès que l'on n'est pas dans ce cas, ou alors on peut écrire, en notant $\alpha_1,\ldots,\alpha_p$ les coefficients précédents,

$$\begin{aligned}\frac{1}{n}I_{\mathcal{V}} &= \sum_{k=1}^{d}\lambda_k\alpha_k + \sum_{k=d+1}^{p}\lambda_k\alpha_k \\ &\leq \sum_{k=1}^{d}\lambda_k\alpha_k + \lambda_{d+1}\sum_{k=d+1}^{p}\alpha_k = \sum_{k=1}^{d}\lambda_k\alpha_k + \lambda_{d+1}\left(d-\sum_{k=1}^{d}\alpha_k\right) = \sum_{k=1}^{d}(\lambda_k-\lambda_{d+1})\alpha_k + d\lambda_{d+1} \\ &\leq \sum_{k=1}^{d}(\lambda_k-\lambda_{d+1}) + d\lambda_{d+1} = \sum_{k=1}^{d}\lambda_k = \frac{1}{n}I_{\mathcal{V}_d}.\end{aligned}$$

Ceci démontre la propriété énoncée : $\mathcal{V}_d = \overline{a} + \mathrm{Vect}(\vec{v}_1,\ldots,\vec{v}_d)$.

En particulier, on observe donc que les sous-espaces maximaux sont emboîtés : $\{\overline{a}\} = \mathcal{V}_0 \subset \mathcal{V}_1 \subset \cdots \subset \mathcal{V}_p = \mathbb{R}^p$.

Et que l'on pourrait construire ces espaces de façon itérative : pour tout d, $\mathbb{R}\vec{v}_{d+1}$ est la droite qui approche le mieux les projections de $a_1,\ldots,a_n$ sur l'orthogonal de V_d. En effet, $\vec{v}_{d+1},\ldots,\vec{v}_p$ engendrent $(V_d)^\perp$ et sont les vecteurs propres de la matrice de covariance de ces projections (écrites dans une base de $(V_d)^\perp$).

Ainsi, $\overline{a}+\mathbb{R}\vec{v}_1$ approche au mieux $a_1,\ldots,a_n$, puis $\mathbb{R}\vec{v}_2$ approche au mieux les "restes" $a_i - \mathrm{Proj}_{\mathbb{R}\vec{v}_1}(a_i)$, puis $\mathbb{R}\vec{v}_3$ approche au mieux $a_i - \mathrm{Proj}_{\mathcal{V}_1}(a_i)$, etc.

2 Éléments d'aide à l'analyse

Il s'agit ensuite d'interpréter ces résultats géométriques en termes des individus et des variables.

Les directions principales $\vec{v}_1,\ldots,\vec{v}_p$ correspondent à de nouvelles variables. En effet, les points a_i ont pour coordonnées les variables "initiales" $x_1,\ldots,x_p$ quand on les écrit dans le repère canonique $(0;\vec{e}_1,\ldots,\vec{e}_p)$ de $\mathbb{R}^p$, et peuvent maintenant être exprimés dans le repère orthonormé $(\overline{a};\vec{v}_1,\ldots,\vec{v}_p)$; les coordonnées deviennent alors de nouvelles variables $c_1,\ldots,c_p$ appelées **1re composante principale**, **2e composante principale**, etc. Vu l'orthonormalité, elles se calculent par produit scalaire :

$$(c_j)_i = (a_i-\overline{a})\cdot\vec{v}_j$$

Notons que c_j a pour variance $\vec{v}_j\Gamma{}^t\vec{v}_j = \lambda_j$.

Ces nouvelles variables ont l'avantage de bien résumer l'information : vu le choix de $\vec{v}_1$, c_1 est la variable de variance maximale et telle que les données sont le mieux approchées si on ne garde que cette variable, puis

de même pour c_1, c_2 parmi les couples de 2 variables, etc. Pour choisir combien de variables conserver, on observe la **proportion d'inertie**

$$\frac{\lambda_1 + \cdots + \lambda_i}{\lambda_1 + \cdots + \lambda_p}$$

apportée par les i premières variables. Diverses règles heuristiques sont utilisées ; en représentant graphiquement $\lambda_1, \ldots, \lambda_p$, on observe souvent qu'un petit nombre de valeurs propres se "détachent" des autres, dont la valeur décroît ensuite doucement ("règle du coude") ; alternativement, on peut fixer un seuil (75 % par exemple) pour la part d'inertie que l'on souhaite conserver.

Ces nouvelles variables ont aussi l'avantage d'être orthogonales : $\mathrm{Cov}(c_j, c_k) = v_j \Gamma^t \vec{v}_k = 0$. Ceci facilite l'interprétation puisqu'il n'y a pas de corrélation entre les composantes. En revanche, ces nouvelles variables n'ont pas d'interprétation intuitive immédiate ; on peut en général les comprendre en observant leur **corrélation** avec les variables initiales, et en observant leurs valeurs sur les données. On calcule la corrélation entre c_k et x_j :

$$\mathrm{Corr}(c_k, x_j) = \frac{\mathrm{Cov}(c_k, x_j)}{\sqrt{\mathrm{Var}(c_k)}\sqrt{\mathrm{Var}(x_j)}} = \frac{\vec{e}_j \Gamma^t \vec{v}_k}{\sqrt{\lambda_k}\sigma(x_j)} = \frac{\sqrt{\lambda_k}}{\sigma(x_j)} \vec{e}_j \cdot \vec{v}_k$$

donc, dans le cas courant où les variables ont été réduites ($\sigma(x_j) = 1$),

$$\mathrm{Corr}(c_k, x_j) = \sqrt{\lambda_k}\, \vec{e}_j \cdot \vec{v}_k.$$

Ainsi, les composantes de $\vec{v}_k$ donnent, au facteur commun $\sqrt{\lambda_k}$ près, les corrélations de c_k avec les variables initiales : il sera utile de représenter graphiquement ces composantes (sur un diagramme en barres) afin d'apprécier lesquelles des variables $x_1, \ldots, x_p$ contribuent le plus, positivement ou négativement, à c_k, et ainsi donner une description intuitive de c_k.

Notons que, pour toute variable x_j, (avec ou sans réduction)

$$\sum_{k=1}^{p} \mathrm{Corr}(x_j, c_k)^2 = \sum_{k=1}^{n} \frac{\lambda_k}{\mathrm{Var}(x_j)} (\vec{e}_j \cdot \vec{v}_k)^2 = \frac{1}{\mathrm{Var}(x_j)} \vec{e}_j \Gamma\, {}^t\vec{e}_j = 1,$$

c'est-à-dire que le point de coordonnées $(\mathrm{Corr}(x_j, c_k))_{1 \le k \le p}$ appartient à la sphère unité de $\mathbb{R}^p$. En pratique on représentera le point $(\mathrm{Corr}(x_j, c_1), \mathrm{Corr}(x_j, c_2))$, pour le comparer au cercle unité ("**cercle des corrélations**") : s'il est proche du cercle unité c'est que la somme ci-dessus est essentiellement donnée par ses deux premiers termes, autrement dit la variable est bien décrite par ses projections sur le premier plan factoriel. On observera aussi quelles variables x_j sont positivement corrélées avec c_1, avec c_2, afin d'interpréter ces variables.

Pour les individus, la qualité de l'approximation par la première composante principale peut se quantifier par le **cosinus** de l'angle θ_{ij} entre $a_i - \overline{a}$ et $\mathbb{R}\vec{v}_j$; il est donné par

$$\cos\theta_{ij} = \frac{(a_i - \overline{a}) \cdot \vec{v}_j}{\|a_i - \overline{a}\|}$$

Ces valeurs fournissent ensuite l'angle $\theta_{i,(1,\ldots,k)}$ entre le vecteur $a_i - \overline{a}$ et le plan engendré par $\vec{v}_1, \ldots, \vec{v}_k$: c'est l'angle entre ce vecteur et sa projection $\sum_{j=1}^{k} (a_i - \overline{a}) \cdot \vec{v}_j\, \vec{v}_j$, ce qui donne, pour $k = 1, \ldots, p$,

$$\cos^2\theta_{i,(1,\ldots,k)} = \frac{\sum_{j=1}^{k} \left((a_i - \overline{a}) \cdot \vec{v}_j\right)^2}{\|a_i - \overline{a}_i\|^2} = \sum_{j=1}^{k} \cos^2\theta_{ij}$$

(et donc en particulier $\sum_{j=1}^{p} \cos^2\theta_{ij} = \cos^2\theta_{i,(1,\ldots,p)} = \cos^2 0 = 1$) Dans un nuage de points des valeurs (c_1, c_2), on peut ainsi par exemple représenter $\cos^2\theta_{i,(1,2)}$ par la couleur des points afin d'apprécier leur proximité à ce plan factoriel, et éventuellement de mettre en évidence des situations où des points lointains ont des projections proches.

3 Un exemple : records en athlétisme par pays

```
ath=read.table("athletics.txt",header=T,row.names="Pays")
rownames(ath)
```

```
##  [1] "Australie"       "Belgique"        "Brésil"          "RoyaumeUni"
##  [5] "Canada"          "Chine"           "Croatie"         "Ethiopie"
##  [9] "France"          "Allemagne"       "Inde"            "Iran"
## [13] "Italie"          "Jamaïque"        "Japon"           "Kenya"
## [17] "Lituanie"        "NouvelleZélande" "Portugal"        "Russie"
## [21] "AfriqueduSud"    "Espagne"         "Suède"           "Suisse"
## [25] "Ukraine"         "USA"
```

```
head(ath)
```

```
##            X100m X200m X400m  X800m X1500m X5000m X10000m SemiMarathon Marathon
## Australie   9.93 20.06 44.38 104.40 211.96 775.76 1649.73         3602     7671
## Belgique   10.02 20.19 44.78 103.86 214.13 769.71 1612.30         3605     7640
## Brésil     10.00 19.89 44.29 101.77 213.25 799.43 1648.12         3573     7565
## RoyaumeUni  9.87 19.87 44.36 101.73 209.67 780.41 1638.14         3609     7633
## Canada      9.84 20.17 44.44 103.68 211.71 793.96 1656.01         3650     7809
## Chine      10.17 20.54 45.25 106.44 216.49 805.14 1670.00         3635     7695
```

```
ath["France","X100m"] # intérêt d'avoir nommé les lignes (et aussi pour les graphes)
```

```
## [1] 9.99
```

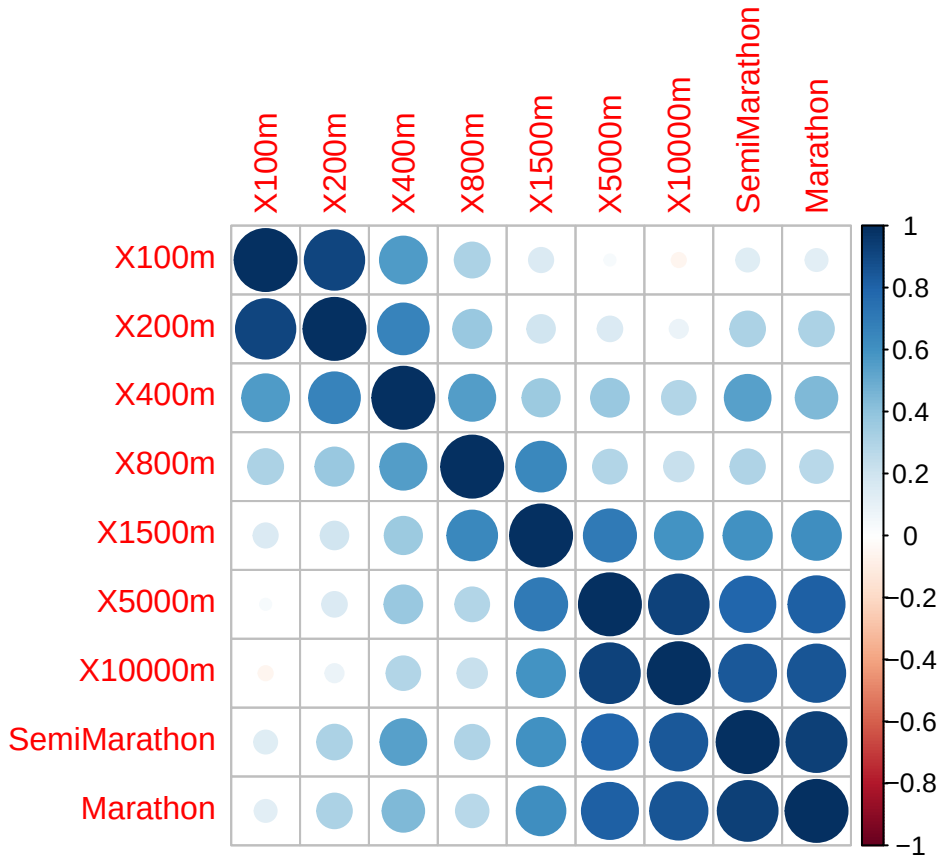
```
# Étude des corrélations
cor(ath)
```

```
##                   X100m     X200m     X400m     X800m    X1500m     X5000m
## X100m         1.00000000 0.9167256 0.5697854 0.3175890 0.1520493 0.03224103
## X200m         0.91672559 1.0000000 0.6698083 0.3778046 0.1985415 0.15825221
## X400m         0.56978539 0.6698083 1.0000000 0.5502815 0.3604686 0.37072209
## X800m         0.31758896 0.3778046 0.5502815 1.0000000 0.6427208 0.29733155
## X1500m        0.15204933 0.1985415 0.3604686 0.6427208 1.0000000 0.70694940
## X5000m        0.03224103 0.1582522 0.3707221 0.2973315 0.7069494 1.00000000
## X10000m      -0.05214076 0.0858235 0.2936163 0.2260386 0.5932397 0.92583521
## SemiMarathon  0.13726468 0.3128569 0.5434661 0.3065523 0.6044177 0.79497017
## Marathon      0.12572109 0.3123263 0.4497096 0.2757651 0.6157718 0.81728219
##                  X10000m SemiMarathon  Marathon
## X100m        -0.05214076    0.1372647 0.1257211
## X200m         0.08582350    0.3128569 0.3123263
## X400m         0.29361634    0.5434661 0.4497096
## X800m         0.22603860    0.3065523 0.2757651
## X1500m        0.59323973    0.6044177 0.6157718
## X5000m        0.92583521    0.7949702 0.8172822
## X10000m       1.00000000    0.8412495 0.8539543
## SemiMarathon  0.84124952    1.0000000 0.9304608
## Marathon      0.85395433    0.9304608 1.0000000
```

```
library(corrplot) # Installé si besoin avec install.packages("corrplot")
```

```
## corrplot 0.92 loaded
```

```
corrplot(cor(ath))
```




```
ACP=prcomp(ath)
ACP$sdev   # écart-types des composantes principales (donc =sqrt(lambda1), etc.)
```

```
## [1] 325.87265960  41.31132383  20.18163270   6.16698108   2.52490922
## [6]   1.08938029   0.48031700   0.30183883   0.06240517
```

```
plot(ACP)  # représente la variance (inertie) de chaque direction principale
```


```
ACP$rotation[,"PC1"]   # donne les coordonnées de la première direction principale
```

```
##        X100m        X200m        X400m        X800m       X1500m       X5000m
## 7.835684e-05 4.454449e-04 1.045464e-03 1.288004e-03 6.950152e-03 4.359417e-02
##      X10000m SemiMarathon     Marathon
## 9.881088e-02 3.478714e-01 9.312734e-01
```

```
summary(ath)
```

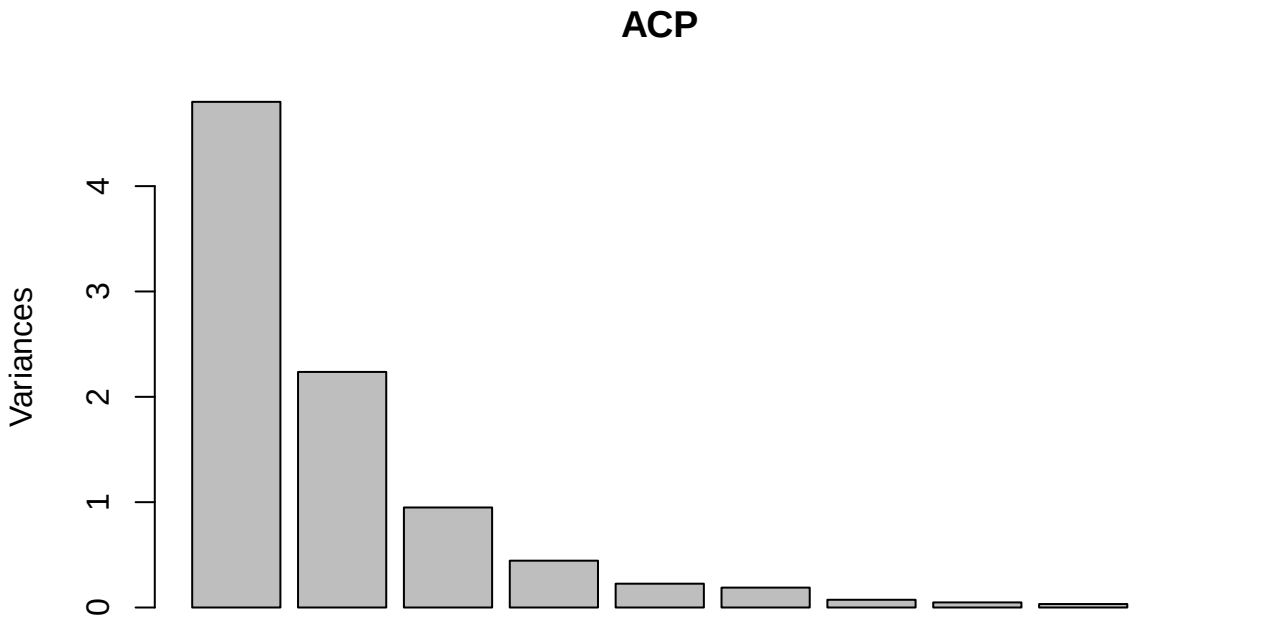
```
##      X100m            X200m           X400m           X800m
##  Min.   : 9.580   Min.   :19.19   Min.   :43.18   Min.   :101.7
##  1st Qu.: 9.992   1st Qu.:20.02   1st Qu.:44.45   1st Qu.:102.8
##  Median :10.065   Median :20.20   Median :44.78   Median :104.0
##  Mean   :10.068   Mean   :20.25   Mean   :44.93   Mean   :104.1
##  3rd Qu.:10.178   3rd Qu.:20.51   3rd Qu.:45.42   3rd Qu.:105.2
##  Max.   :10.500   Max.   :21.11   Max.   :46.37   Max.   :106.6
##      X1500m          X5000m          X10000m      SemiMarathon     Marathon
##  Min.   :206.3   Min.   :757.4   Min.   :1578   Min.   :3513   Min.   :7439
##  1st Qu.:210.5   1st Qu.:779.2   1st Qu.:1637   1st Qu.:3606   1st Qu.:7594
##  Median :212.2   Median :790.5   Median :1651   Median :3652   Median :7642
##  Mean   :213.1   Mean   :790.0   Mean   :1656   Mean   :3673   Mean   :7755
##  3rd Qu.:215.9   3rd Qu.:797.8   3rd Qu.:1673   3rd Qu.:3684   3rd Qu.:7814
##  Max.   :220.9   Max.   :833.4   Max.   :1763   Max.   :4103   Max.   :8903
```

```
apply(ath,2,sd) # calcul de l'écart-type (sd) de chaque variable (cf. TP1)
```

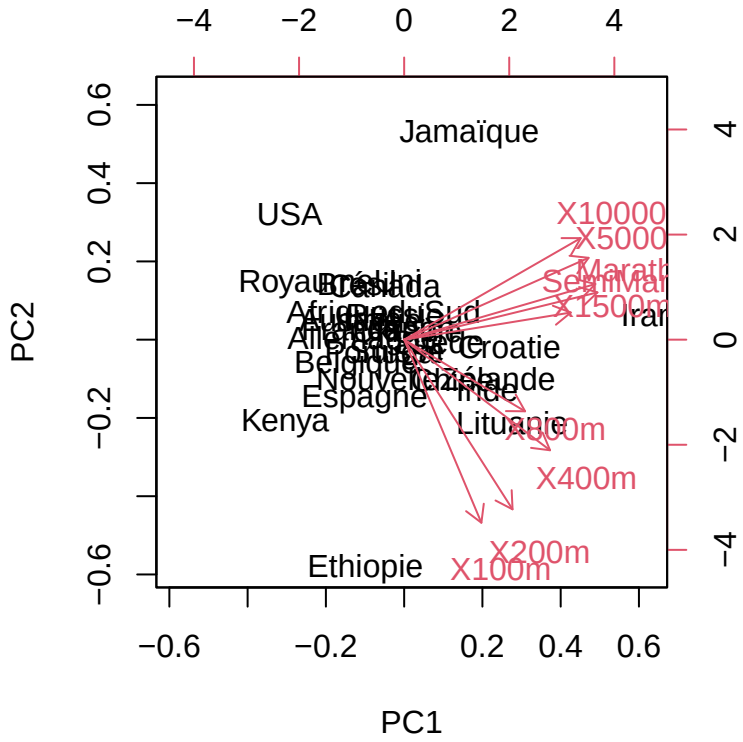
```
##        X100m        X200m        X400m        X800m       X1500m       X5000m
##    0.2023268    0.4644659    0.7340192    1.4888056    3.6510158   17.2388961
##      X10000m SemiMarathon     Marathon
##   37.3334810  119.6924585  303.8408111
```

```
# l'écart-type du marathon et semi-marathon sont prépondérants, donc mécaniquement
# constituent la partie prépondérante (PC1) de la variance. Ce que l'on veut étudier,
# ce sont plutôt les covariances que les variances, mais elles sont ici négligeables
# => on va normer les données.
```

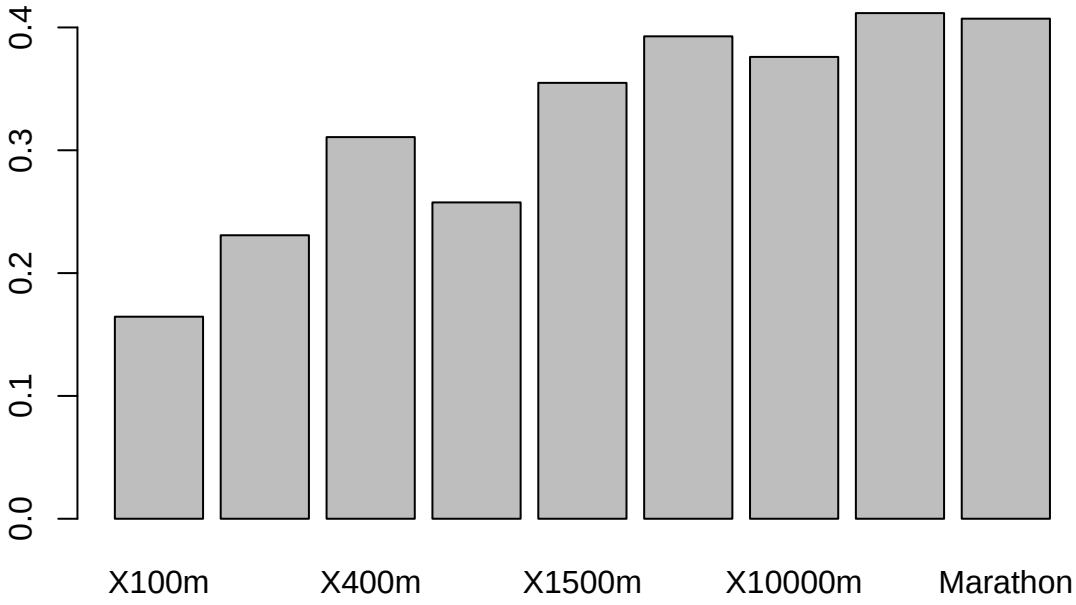
```
ACP=prcomp(ath,scale.=T)
plot(ACP) # ici, les variances sont davantage d'ordres comparables
```



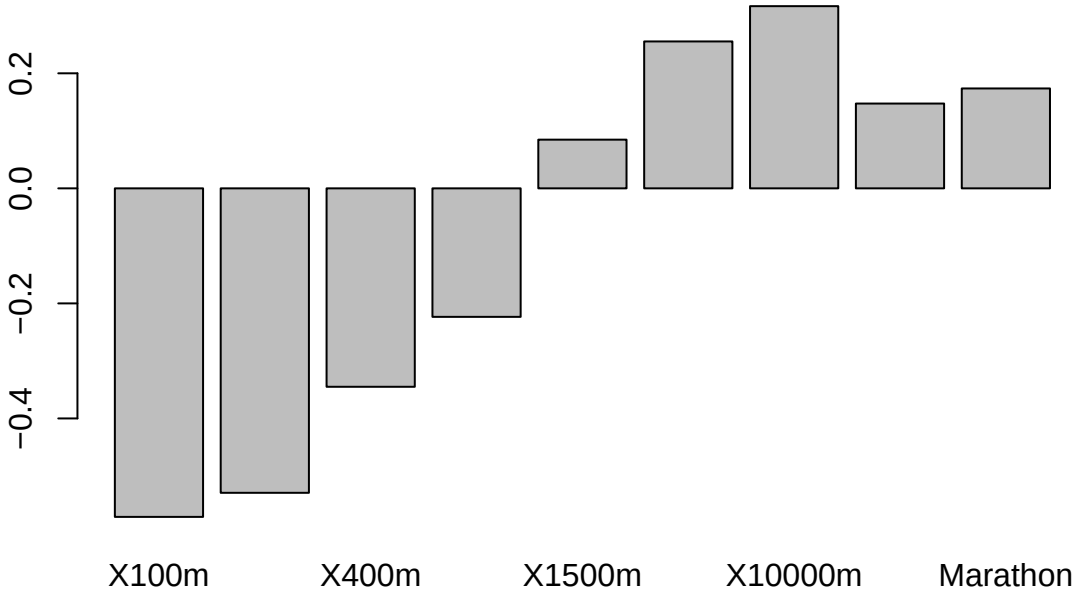

```
biplot(ACP)
```



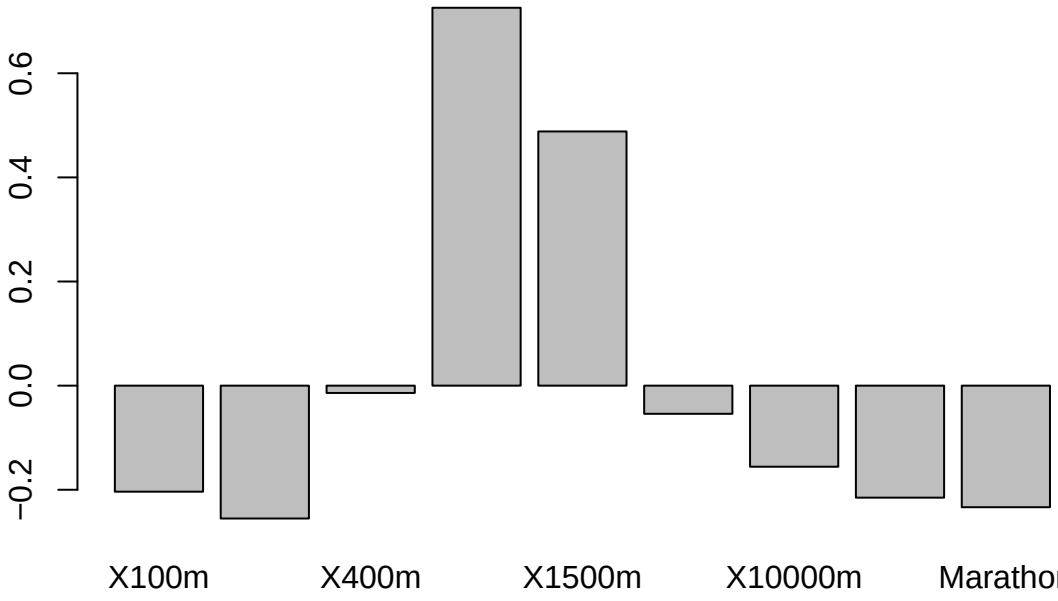

```
barplot(ACP$rotation[,"PC1"])  # représentation graphique des composantes de v_1
```




```
# La première composante principale donne des poids comparables aux disciplines :
# elle mesure en quelque sorte la performance globale
barplot(ACP$rotation[,"PC2"])  # représentation graphique des composantes de v_2
```

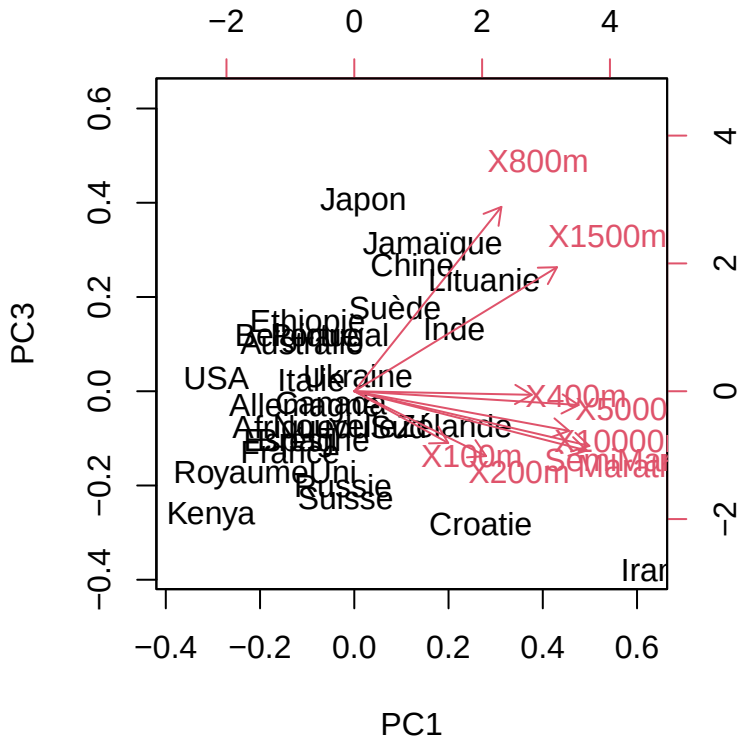



```
# La deuxième composante oppose les performance en sprint et en fond, elle met
# en évidence les pays ayant des performances très différentes dans ces classes
# de disciplines.
barplot(ACP$rotation[,"PC3"])  # représentation graphique des composantes de v_3
```




```
# La troisième composante pondère principalement les distances intermédiaires

# On peut représenter les individus dans le plan Vect(v_1,v_3)
biplot(ACP,choices=c(1,3))
```

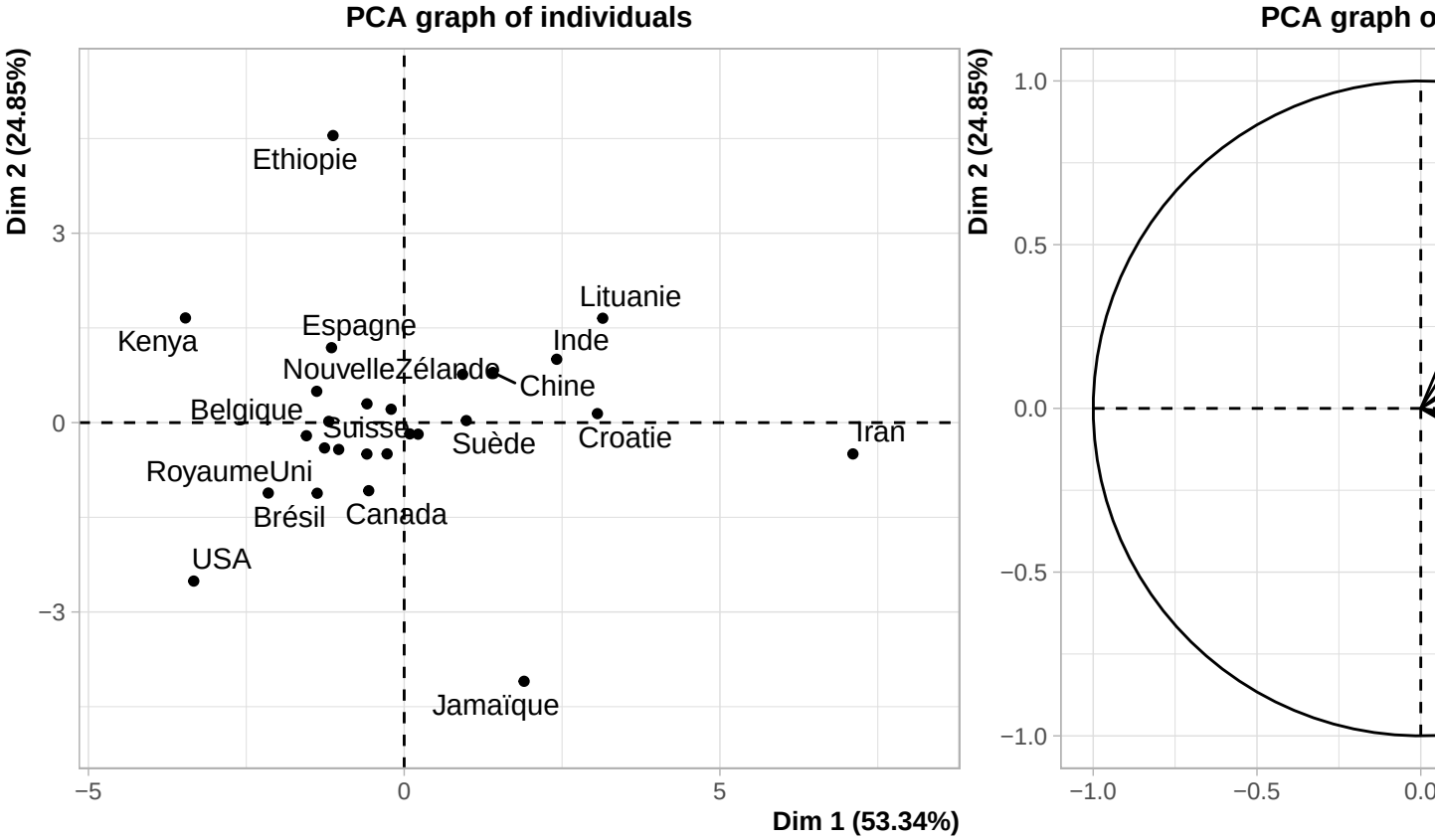


Plusieurs packages ont été développés pour améliorer entre autres le rendu graphique des fonctions standards de R. On recommande en particulier le package “FactoMineR” (développé à l’université de Rennes).

```
# Pour comparer avec le rendu graphique d'un autre package d'ACP
library(FactoMineR)
res.ACP=PCA(ath)
```

```
## Warning: ggrepel: 9 unlabeled data points (too many overlaps). Consider
```

```
## increasing max.overlaps
```




```
summary(res.ACP)
```

```
##
## Call:
## PCA(X = ath)
##
##
## Eigenvalues
##                        Dim.1   Dim.2   Dim.3   Dim.4   Dim.5   Dim.6   Dim.7
## Variance               4.800   2.236   0.950   0.445   0.226   0.189   0.073
## % of var.             53.336  24.847  10.550   4.940   2.511   2.097   0.816
## Cumulative % of var.  53.336  78.182  88.732  93.672  96.184  98.281  99.097
##                        Dim.8   Dim.9
## Variance               0.048   0.033
## % of var.              0.538   0.365
## Cumulative % of var.  99.635 100.000
##
## Individuals (the 10 first)
##                  Dist    Dim.1    ctr   cos2    Dim.2    ctr   cos2    Dim.3
## Australie     |  1.602 | -1.261  1.275  0.620 | -0.401  0.277  0.063 |  0.519
## Belgique      |  1.884 | -1.385  1.537  0.541 |  0.496  0.423  0.069 |  0.571
## Brésil        |  2.357 | -1.377  1.520  0.341 | -1.119  2.154  0.225 | -0.520
## RoyaumeUni    |  2.621 | -2.152  3.712  0.674 | -1.116  2.141  0.181 | -0.901
## Canada        |  1.475 | -0.562  0.253  0.145 | -1.078  2.000  0.534 | -0.125
## Chine         |  2.336 |  1.405  1.581  0.362 |  0.795  1.086  0.116 |  1.359
## Croatie       |  3.430 |  3.060  7.503  0.796 |  0.143  0.035  0.002 | -1.421
```

```
## Ethiopie     |  4.774 | -1.127  1.018  0.056 |  4.548 35.572  0.907 |  0.726
## France       |  1.795 | -1.548  1.921  0.744 | -0.208  0.074  0.013 | -0.652
## Allemagne    |  1.439 | -1.194  1.143  0.689 |  0.020  0.001  0.000 | -0.181
##                 ctr   cos2
## Australie     1.091  0.105 |
## Belgique      1.320  0.092 |
## Brésil        1.096  0.049 |
## RoyaumeUni    3.287  0.118 |
## Canada        0.063  0.007 |
## Chine         7.484  0.339 |
## Croatie       8.177  0.172 |
## Ethiopie      2.133  0.023 |
## France        1.723  0.132 |
## Allemagne     0.132  0.016 |
##
## Variables
##                 Dim.1    ctr   cos2    Dim.2    ctr   cos2    Dim.3    ctr
## X100m        |  0.360  2.706  0.130 |  0.854 32.625  0.730 | -0.198  4.148
## X200m        |  0.506  5.327  0.256 |  0.792 28.017  0.627 | -0.249  6.513
## X400m        |  0.681  9.654  0.463 |  0.516 11.903  0.266 | -0.014  0.020
## X800m        |  0.564  6.633  0.318 |  0.334  4.992  0.112 |  0.707 52.681
## X1500m       |  0.778 12.594  0.605 | -0.127  0.717  0.016 |  0.476 23.839
## X5000m       |  0.861 15.426  0.740 | -0.382  6.524  0.146 | -0.053  0.292
## X10000m      |  0.824 14.137  0.679 | -0.474 10.034  0.224 | -0.152  2.427
## SemiMarathon |  0.902 16.948  0.814 | -0.220  2.174  0.049 | -0.210  4.628
## Marathon     |  0.892 16.575  0.796 | -0.260  3.016  0.067 | -0.228  5.453
##                cos2
## X100m         0.039 |
## X200m         0.062 |
## X400m         0.000 |
## X800m         0.500 |
## X1500m        0.226 |
## X5000m        0.003 |
## X10000m       0.023 |
## SemiMarathon  0.044 |
## Marathon      0.052 |
```

4 Exercice : ACP "à la main"

Afin de bien comprendre le fonctionnement de l'ACP, on réécrit les calculs menant à l'ACP des données "athletics" du cours.

1. Chargez les données du fichier "athletics.csv" dans un dataframe nommé `ath`.
2. Calculez le vecteur des écarts-types des variables (cf. `apply` dans le cours n°1), l'enregistrer dans une variable `sigma_ath`. De même pour les moyennes, dans une variable `mean_ath`.
3. Définir un second data frame `ath_norm` qui contient une version centrée réduite du data frame précédent. On pourra faire une boucle sur les colonnes de `ath`. (Ou alors il y a la fonction `scale`)
4. Par un produit matriciel, obtenir la matrice des covariances `G` (de `ath_norm`), vérifier que les variances sont égales à 1. *On pourra utiliser `as.matrix` pour convertir un data.frame en matrice.*
5. Calculer le vecteur `sdev` des valeurs propres et la matrice `rotation` des vecteurs propres de `G`. *Utiliser `eigen` (qui trie les valeurs propres par ordre décroissant).* Pour faciliter l'utilisation, donner des noms aux lignes et colonnes de `rotation` (lignes : `rownames(ath)`, colonnes : `paste("PC",1:9,sep="")`)
6. Ajouter à `ath_norm` une variable `PC1` qui donne la composante principale de chaque observation, et de même pour `PC2`. C'est donc le produit scalaire avec le premier vecteur propre (première colonne de

`rotation`), et le deuxième.

7. Représenter graphiquement les observations, par leur nom, dans le premier plan factoriel (plan des deux premières composantes principales). *On pourra utiliser `+geom_text()` avec l'esthétique `label=rownames(ath)`, et éventuellement préciser `color=rownames(ath)` et (hors de `aes`) `show.legend=F` pour assigner des couleurs différentes aux étiquettes.*
8. Représenter graphiquement le cercle des corrélations des variables avec les deux premières composantes principales. Autrement dit, représenter pour chaque variable un point ayant pour abscisse la corrélation entre cette variable et `PC1`, et pour ordonnée la corrélation entre cette variable et `PC2`. C'est aussi le point de coordonnées égales, pour la i-ième variable, à $(\sqrt{\lambda_1}(v_1)_i, \sqrt{\lambda_2}(v_2)_i)$ où λ_1 est la plus grande valeur propre, associée à la direction principale v_1 (et $(v_1)_i$ est la i-ième composante de v_1, c'est-à-dire `rotation[i,1]` avec la notation de la q.4). Optionnel : représenter également des flèches comme dans `biplot`, en utilisant `geom_segment`, et aussi le cercle de centre 0 et de rayon 1.
9. Comparer aux résultats de `PCA=prcomp(ath,scale=T)` et `biplot(PCA)`.

5 Exercice : Autre exemple sportif

Étudions les données "decathlon" par ACP.

1. Charger les données. Regarder ce dont il s'agit. On ne retient que les résultats aux 10 épreuves : définir un dataframe `decathlon` restreint à ces variables, et traduire en français (bref) le nom des variables (éditer `colnames(decathlon)=c("X100m","Saut.Long",...)`).
2. Réaliser l'ACP normée sur ces données.
3. Représenter les variances (ou "inerties") des axes factoriels.
4. Représenter les observations selon le premier plan factoriel, et les corrélations des variables avec les 2 premiers axes factoriels.
5. Calculer la part d'inertie représentée par les 2 premiers axes.
6. D'après le graphe dans le 1er plan factoriel, que dire de

— Bourguignon et Karpov ?
— Barras et Qi ?
— Casarsa ?
— Serble et Clay ?

7. Comment interpréter les 2 premiers composantes principales ?
8. Que dire des scores en 100m et 110m haie ? Et quel lien avec le saut en longueur ?
9. Que peut-on dire des scores en javelot ou saut à la perche ?
10. Qu'est ce que le 3ème axe factoriel semble représenter ?
11. Combien d'axes factoriels conserveriez vous ?

6 Exercice : Températures en Europe

On s'intéresse à des données de température de villes d'Europe. Pour un meilleur rendu graphique, on conseille d'utiliser le package FactoMineR ou vos propres fonctions écrites au premier exercice.

1. Ouvrir `temperatures.csv` et observer les données : pouvez-vous deviner ce que sont les variables ?
2. Charger ces données dans un data frame, en donnant pour nom de lignes les noms de villes.
3. Réaliser l'ACP normée (c'est-à-dire en centrant et standardisant les variables) des 12 variables de température mensuelles.
4. Représenter les variances (ou inerties) des axes factoriels. Combien d'axes semble-t-il raisonnable de conserver ? Calculer la part d'inertie représentée par les 2 premières directions principales.
5. Représenter les observations selon le premier plan factoriel, et les corrélations des variables avec les 2 premiers axes factoriels. D'après le second graphe, semble-t-il que les variables sont bien représentées par les deux premières composantes principales ?

6. Représenter également des diagrammes en bâtons des composantes des deux premières directions principales. Comment peut-on interpréter ces deux composantes ?
7. Sur le graphe des individus, peut-on identifier des points communs géographiques entre les villes dont les projections sont proches ? Si vous utilisez votre propre fonction, changez la coloration (dans `ggplot`) pour qu'elle soit donnée par la variable `Region` pour confirmer votre impression.
8. Avec FactoMineR, on peut ajouter des variables (ou des individus) non prises en compte dans le calcul de l'ACP mais que l'on souhaite représenter graphiquement ; on peut l'utiliser ici pour représenter la région, la latitude et la longitude. Utiliser la syntaxe `plot(PCA(data,quali.sup=v,quanti.sup=w),habillage=c)` où `data` est le data frame contenant les 15 variables du jeu de données initial, où `v` est le vecteur des numéros de colonnes qualitatives non prises en compte dans l'ACP, `w` est le vecteur des numéros de colonnes quantitatives non prises en compte dans l'ACP, et `c` est la colonne de la variable (ici Région) à utiliser pour colorer les points. Commenter les corrélations de la longitude et de la latitude avec les deux premières composantes principales.
9. Ajouter au data frame initial deux variables "Moyenne" et "Amplitude" égales à la moyenne annuelle des températures, et à l'écart maximal entre températures dans l'année. (Pour l'amplitude, on pourra utiliser une boucle `for(i in 1:n){ ... }`). Les ajouter au graphe de la question précédente et commenter.