Modélisation en biologie.

Organisation de l'UE

- 5 séances le lundi après-midi du 23 janvier au 20 février
- 2 séances au mois de mars
- MCC : 0.5EE + 0.2CC + 0.3CCTP
- Le TP noté aura lieu en séance 6.
- Le CC sera un contrôle de connaissance et aura lieu en séance 7.
- La date de l'examen n'est pas encore fixée.

Rappel sur quelques lois de probabilités usuelles en modélisation

On rappelle ici 4 lois usuelles très connus et récurentes dans la modélisation des données en biologie, mais cette liste est loin d'être exhaustive. Pour aller plus loin, vous pouvez consulter le lien suivant : `http://www.math.wm.edu/~leemis/chart/UDR/UDR.html`.

Loi de Bernoulli

Modélisation

- La Loi de Bernoulli modélise une expérience aléatoire à 2 issues possibles.
- Les exemples sont multiples :
  - le lancé d'une pièce de monnaie (pile ou face)
  - le succès à un jeu (gagné ou perdu)
  - la mutation d'un nucléotide (muté ou non muté)
  - l'interaction entre deux protéines (présence ou absence de l'interaction)

Définition

- Soit une variable aléatoire X suivant une loi de Bernoulli. On encode les deux issues possibles par les valeurs 0 et 1.
- On note $p = P(X = 1)$ la probabilité de "succès". Ici, $p \in [0,1]$.
- Le support de $X \sim \mathscr{B}(p)$ est $X(\Omega) = \{0,1\}$.
- La loi de X est donnée par $P(X = k) = p^k(1-p)^{1-k}$ pour tout $k \in \{0,1\}$.

Propriétés

Soit $X \sim \mathscr{B}(p)$:
- $E(X) = p$
- $Var(X) = p(1-p)$
- $\mathscr{M}_X(t) = E(e^{tX}) = (1-p) + pe^t$

La loi de Bernoulli sous R

- Pour simuler une expérience de Bernoulli, il existe la fonction 'sample' :

```
sample(c(0,1),1,prob=c(1/3,2/3))
## [1] 1
```

— On peut également utiliser la fonction ‘rbinom‘ :

```
rbinom(1,1,prob=2/3)
## [1] 1
```

— La fonction ‘dbinom()‘ permet d’obtenir la loi de la loi de Bernoulli :
  — Soit $X \sim \mathscr{B}(p=2/3)$, la valeur $P(X=0)$ est donnée par :

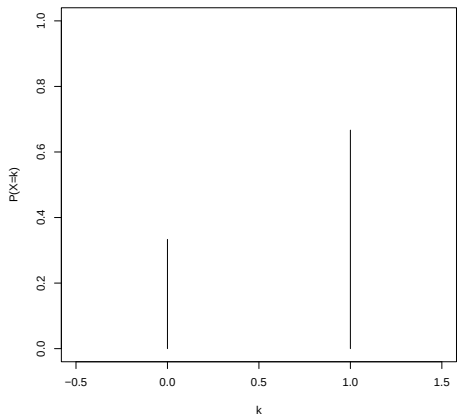
```
dbinom(0,1,prob=2/3)
## [1] 0.3333333
```

  — Soit $X \sim \mathscr{B}(p=2/3)$, la valeur $P(X=1)$ est donnée par :

```
dbinom(1,1,prob=2/3)
## [1] 0.6666667
```

  — On peut obtenir une représentation graphique de la loi de la loi de Bernoulli $X \sim \mathscr{B}(p=2/3)$:

```
plot(0:1, dbinom(0:1,1,prob=2/3),
     type="h",
     ylab="P(X=k)",
     xlab="k",
     ylim=c(0,1),
     xlim=c(-0.5,1.5))
```



Loi Binomiale

Définition

— Soit une variable aléatoire Y suivant une loi Binomiale de paramètres n et p, notée $\mathscr{B}(n,p)$.
— Le support de Y est $Y(\Omega)=\{0,\dots,n\}$.
— La loi de Y est donnée par, pour tout $k \in Y(\Omega)$

$$P(Y=k)=\binom{n}{k}p^k(1-p)^{n-k}$$

— Rappel : Le coefficient $\binom{n}{k}$ vaut $\frac{n!}{k!(n-k)!}$ est appelé le coefficient binomiale. Il représente également le nombre de combinaisons de k objets parmi n. Ses valeurs sont données par la n-ième ligne du triangle de Pascal.

Propriétés

Soit $X \sim \mathscr{B}(n,p)$:
— $E(X)=np$
— $Var(X)=np(1-p)$
— $\mathscr{M}_X(t)=E(e^{tX})=((1-p)+pe^t)^n$

Modélisations

— Si l'on réalise $n \in \mathbb{N}^\star$ expériences de Bernoulli $X_1, \ldots, X_n$, chaque X_i suivant une loi $\mathscr{B}(p)$ de paramètre $p \in [0;1]$ et que l'on s'intéresse à la somme de ces variables aléatoires Y alors on obtient une variable aléatoire suivant une loi binomiale de paramètres n et p :

$$X_i \sim \mathscr{B}(p), i \in \{1, \ldots, n\}$$

$$Y = X_1 + \ldots + X_n$$

$$Y \sim \mathscr{B}(n, p)$$

— Remarque : La v.a. $Y \sim \mathscr{B}(n = 1, p)$ n'est autre qu'une loi de Bernoulli $\mathscr{B}(p)$.
— La loi Binomiale permet de modéliser facilement le nombre de succès à un jeu de hasard, le nombre de mutations dans un génome, le nombre d'interactions entre protéines... tout expérience où l'on compte un nombre du succès dans une expérience aléatoire de Bernoulli.

La loi binomiale sous R

— Pour simuler un vecteur de 10 réalisations d'une v.a. de Bernoulli de paramètre $\mathscr{B}(p = \frac{1}{3})$, on utilise :

```
rbinom(n=10,size=1,prob=1/3)
##  [1] 1 0 1 0 0 0 1 0 1 1
```

— Pour simuler une réalisation d'une v.a. binomiale $\mathscr{B}(n = 10; p = \frac{1}{3})$, on utilise

```
rbinom(n=1,size=10,prob=1/3)
## [1] 4
```

— Pour simuler un vecteur de 15 réalisations d'une v.a. binomiale $\mathscr{B}(n = 10; p = \frac{1}{3})$, on utilise :

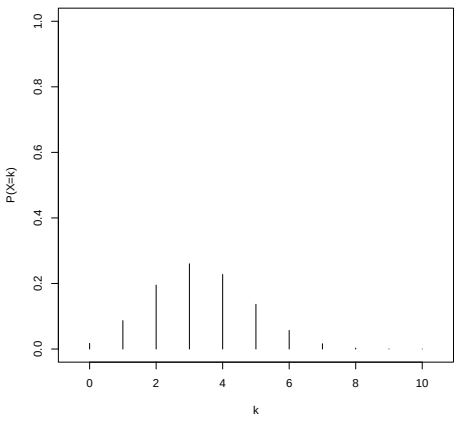
```
rbinom(n=15,size=10,prob=1/3)
##  [1] 3 4 3 5 0 4 2 6 4 6 2 3 3 3 3
```

— NB : le paramètre 'size' de la fonction 'rbinom()' correspond au paramètre n dans $\mathscr{B}(n; p)$.
— La fonction 'dbinom()' permet d'obtenir les valeurs de la loi Binomiale $Y \sim \mathscr{B}(n; p)$:
  — La loi de $Y \sim \mathscr{B}(n = 10; p = \frac{1}{3})$ est donnée par :

```
dbinom(0:10,10,prob=1/3)
##  [1] 1.734153e-02 8.670765e-02 1.950922e-01 2.601229e-01 2.276076e-01
##  [6] 1.365645e-01 5.690190e-02 1.625768e-02 3.048316e-03 3.387018e-04
## [11] 1.693509e-05
```

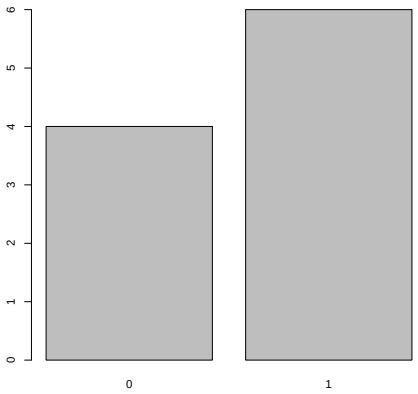
  — Sa représentation graphique est donnée par :

```
plot(0:10, dbinom(0:10,10,prob=1/3),type="h",ylab="P(X=k)", xlab="k",ylim=c(0,1),
```



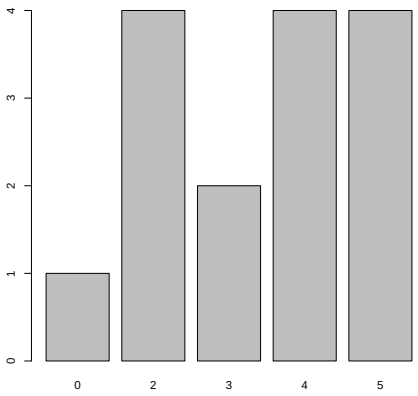
— Simulation d'un vecteur de 10 réalisations d'une v.a. de Bernoulli de paramètre $\mathscr{B}(p = \frac{1}{3})$.

```
vect <- rbinom(n=10,size=1,prob=1/3)
vect
##  [1] 1 1 1 1 0 0 0 1 0 1
barplot(table(vect))
```



— Simulation d'un vecteur de 15 réalisations d'une v.a. binomiale $\mathscr{B}(n=10; p=\frac{1}{3})$.

```
vect2 <- rbinom(n=15,size=10,prob=1/3)
vect2
##  [1] 0 4 2 5 2 3 2 5 3 4 4 2 5 5 4
barplot(table(vect2))
```



Loi de Poisson

Définition

— Soit une variable aléatoire Y suivant une loi de Poisson de paramètre λ, notée $\mathscr{P}(\lambda)$.
— Le support de Y est $Y(\Omega)=\mathbb{N}$.
— La loi de Y est donnée par, pour tout $k \in Y(\Omega)$

$$P(Y=k)=\exp(-\lambda)\frac{\lambda^k}{k!}$$

Propriétés

Soit $X \sim \mathscr{P}(\lambda)$:
— $E(X)=\lambda$
— $Var(X)=\lambda$
— $\mathscr{M}_X(t)=E(e^{tX})=\exp(\lambda(e^t-1))$

Modélisations

— La loi de Poisson modélise généralement un décompte d'évènements rares. Par exemple, on considère un jeu de hasard où la probabilité de gagner est faible $p = 0.01$ mais le nombre d'essais est grand $n = 100$. Dans ce cas, gagner est un évenement rare et l'on peut dire que la loi de la v.a. $Y \sim \mathscr{B}(n = 100; p = 0.01)$ peut être approchée par une loi de Poisson de paramètre $\mathscr{P}(\lambda = n \times p = 1)$.
— La loi de Poisson ne possède qu'un seul paramètre λ, contrairement à la loi Binomiale qui possède deux paramètres. Ce paramètre λ correspond au nombre d'événements moyen observé, l'espérance mathématique de la variable aléatoires Y.

La loi de Poisson sous R

— Pour obtenir la loi d'une loi de Poisson, on utilise la fonction `dpois` :

```
dpois(0:4,lambda=1)
## [1] 0.36787944 0.36787944 0.18393972 0.06131324 0.01532831
```

— Pour simuler des réalisations d'une v.a. suivant une loi de Poisson, on utilise la fonction `rpois` :

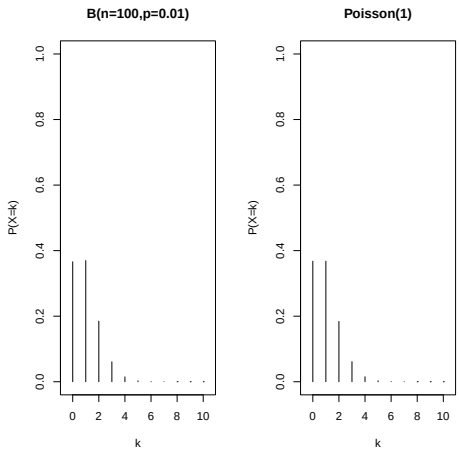
```
rpois(3,lambda=1)
## [1] 0 1 0
```

— Comparaison entre la loi de Poisson $\mathscr{P}(\lambda = 1)$ et la loi Binomiale $\mathscr{B}(n = 100; p = 0.01)$

```
0:4
## [1] 0 1 2 3 4
dbinom(0:4,100,prob=0.01)
## [1] 0.36603234 0.36972964 0.18486482 0.06099917 0.01494171
dpois(0:4,lambda=1)
## [1] 0.36787944 0.36787944 0.18393972 0.06131324 0.01532831
```

— Les représentations graphiques des lois de $Y \sim \mathscr{B}(n = 100; p = 0.01)$ et $Y \sim \mathscr{P}(\lambda = 1)$ sont :

```
par(mfrow=c(1,2))
plot(0:10, dbinom(0:10,100,prob=0.01),
     type="h",ylab="P(X=k)",
     xlab="k",ylim=c(0,1),
     xlim=c(-0.5,10.5),
     main="B(n=100,p=0.01)")
plot(0:10, dpois(0:10,lambda=1),
     type="h",
     ylab="P(X=k)",
     xlab="k",ylim=c(0,1),
     xlim=c(-0.5,10.5),
     main="Poisson(1)")
```



Les représentations graphiques des deux lois $\mathscr{B}(n = 100; p = 0.01)$ et $\mathscr{P}(\lambda = 1)$ sont très similaires.

La loi normale

Définition

— Soit X une v.a. suivant une loi normale $X \sim \mathcal{N}(\mu, \sigma^2)$ de moyenne μ et σ^2 :

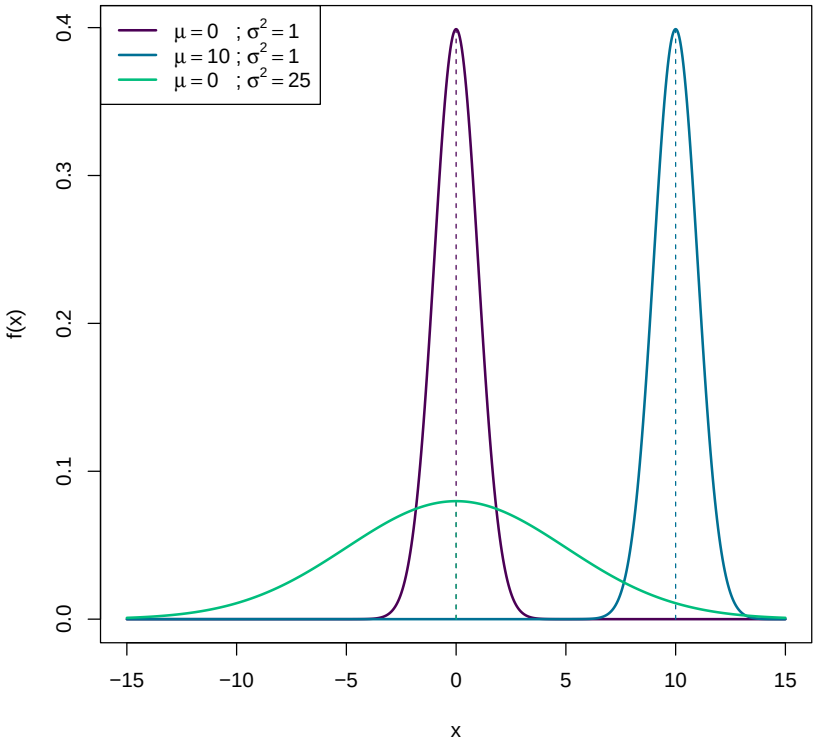
$$X(\Omega) = \mathbb{R}$$

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Propriétés

Soit $X \sim \mathcal{N}(\mu, \sigma^2)$:

— $E(X) = \mu$
— $Var(X) = \sigma^2$
— $\mathcal{M}_X(t) = \exp(\mu t + \frac{\sigma^2 t^2}{2})$
— Si $\lambda > 15$, alors les distributions d'une loi de Poisson $\mathcal{P}(\lambda)$ et d'une loi normale $\mathcal{N}(\lambda, \lambda)$ sont très proches.
— Si $n > 30$, $np > 5$ et $n(1-p) > 5$ alors les distributions d'une loi binomiale $\mathcal{B}(n, p)$ et d'une loi normale $\mathcal{N}(np, np(1-p))$ sont très proches.
— Si $X \sim \mathcal{N}(\mu, \sigma^2)$ alors $X^2 \sim \chi_2(1)$ (preuve à faire en utilisant les fonctions génératrices des moments).



La loi normale sous R

— La fonction 'dnorm' donne la distribution de la loi sous R. Par exemple, $X \sim \mathcal{N}(\mu = 3, \sigma^2 = 2)$, on obtient la valeur de la densité au point $x = 1$ en utilisant la commande :

```
dnorm(1,mean=3,sd=sqrt(2))
## [1] 0.1037769
```

— Soit X une v.a. suivant une loi normale $X \sim \mathcal{N}(\mu = 0, \sigma^2 = 1)$. Pour représenter la densité d'une loi continue, on prend une grille de valeur dans le support de la loi :

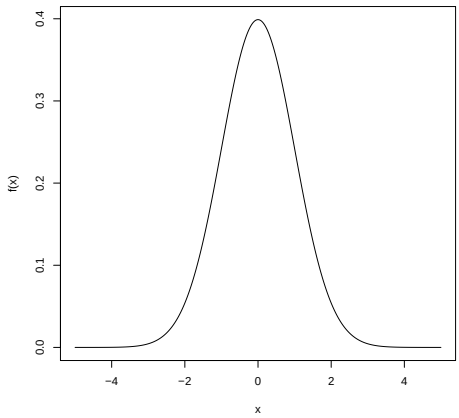
```
grille <- seq(-5,5,length.out=10)
grille
##  [1] -5.0000000 -3.8888889 -2.7777778 -1.6666667 -0.5555556  0.5555556
##  [7]  1.6666667  2.7777778  3.8888889  5.0000000
```

On évalue la fonction f sur chaque élement de la grille

```
dnorm(grille,mean=0,sd=1)
##  [1] 1.486720e-06 2.074403e-04 8.421534e-03 9.947714e-02 3.418923e-01
##  [6] 3.418923e-01 9.947714e-02 8.421534e-03 2.074403e-04 1.486720e-06
```

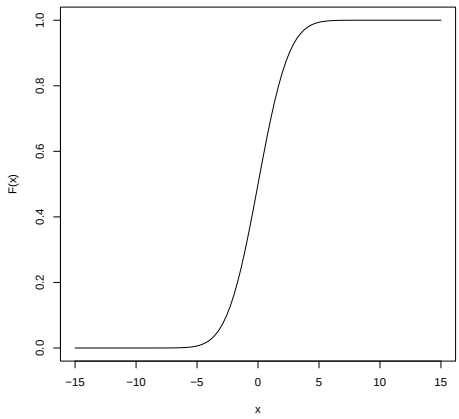
Représentation graphique de la densité de la loi normale $X \sim \mathcal{N}(\mu = 0, \sigma^2 = 1)$

```
grille <- seq(-5,5,length.out=1000)
plot(grille,dnorm(grille,mean=0,sd=1),type="l",xlab="x",ylab="f(x)")
```



— La fonction de répartition de $X \sim \mathcal{N}(\mu = 0, \sigma^2 = 4)$

```
grille=seq(-15,15,length.out=100)
plot(grille,pnorm(grille,mean=0,sd=2),type="l",ylab="F(x)",xlab="x")
```



— Si $X \sim \mathcal{N}(0,1)$, le quantile d'ordre $p = 0.90$ est donné par la commande suivante :

```
qnorm(0.90,mean=0,sd=1)
## [1] 1.281552
```

— Si $X \sim \mathcal{N}(0,1)$, on peut utiliser la fonction `pnorm` pour calculer $P(X \leq 2)$

```
pnorm(2,0,1)
## [1] 0.9772499
```

La loi normale multivariée

Définition

— Soit $\mathbf{X}$ une v.a. suivant une loi normale multivariée si $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ de moyenne $\boldsymbol{\mu}$ et de matrice de variance covariance $\boldsymbol{\Sigma}$:

$$\boldsymbol{\mu} = \mathrm{E}[\mathbf{X}] = (\mathrm{E}[X_1], \mathrm{E}[X_2], \ldots, \mathrm{E}[X_k])^{\mathbf{T}}$$
$$\Sigma_{i,j} = \mathrm{E}[(X_i - \mu_i)(X_j - \mu_j)] = \mathrm{Cov}[X_i, X_j]$$

— La matrice $\boldsymbol{\Sigma}$ est une matrice de variance covariance, symétrique et à coefficients réels.

— La densité du vecteur $\mathbf{X}$ est :

$$f_{\mathbf{X}}(x_1, \ldots, x_k) = \frac{\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)}{\sqrt{(2\pi)^k|\boldsymbol{\Sigma}|}}$$

— Cas de la loi normale bivariée (X, Y) : Soit le coefficient de Pearson $\rho_{X,Y} = \frac{\mathrm{cov}(X,Y)}{\sigma_X \sigma_Y}$. La densité du vecteur aléatoire (X, Y) est :

$$f(x,y) = \frac{1}{2\pi\sigma_X\sigma_Y\sqrt{1-\rho^2}} \exp\left(-\frac{1}{2(1-\rho^2)}\left[\left(\frac{x-\mu_X}{\sigma_X}\right)^2 - 2\rho\left(\frac{x-\mu_X}{\sigma_X}\right)\left(\frac{y-\mu_Y}{\sigma_Y}\right) + \left(\frac{y-\mu_Y}{\sigma_Y}\right)^2\right]\right)$$

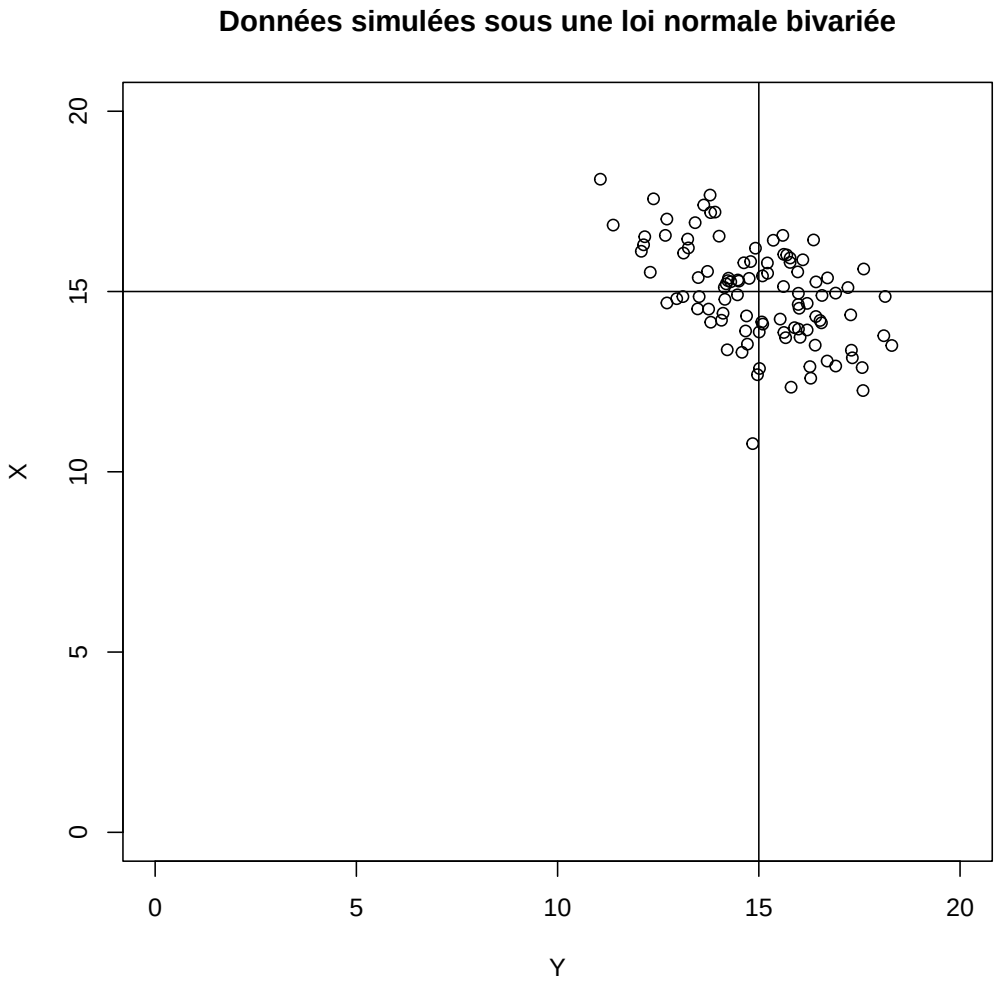
$$\boldsymbol{\mu} = \begin{pmatrix}\mu_X \\ \mu_Y\end{pmatrix}, \quad \boldsymbol{\Sigma} = \begin{pmatrix}\sigma_X^2 & \rho\sigma_X\sigma_Y \\ \rho\sigma_X\sigma_Y & \sigma_Y^2\end{pmatrix}.$$

La loi normale multivariée sous R

— La simulation de données sous une loi normale multivariée nécessite l'installation du package `mvtnorm`.

— Simulation de données sous une loi normale de moyenne $\boldsymbol{\mu} = \begin{pmatrix}15 \\ 15\end{pmatrix}$, $\boldsymbol{\Sigma} = \begin{pmatrix}2 & -1 \\ -1 & 2\end{pmatrix}$

```
library(mvtnorm)
sigmaA=matrix(c(2,-1,-1,2),ncol=2)
data <- rmvnorm(100,mean =  c(15,15),sigma=sigmaA)
plot(data[,1],data[,2],ylab="X",xlab="Y",
     main="Données simulées sous une loi normale bivariée",
     ylim=c(0,20),xlim=c(0,20))
abline(v=15)
abline(h=15)
```



— La décomposition spectrale de la matrice de variance covariance est informative.

```
eigen(sigmaA)
## eigen() decomposition
## $values
## [1] 3 1
##
## $vectors
##            [,1]       [,2]
## [1,] -0.7071068 -0.7071068
## [2,]  0.7071068 -0.7071068
```

Introduction à l'analyse en composantes principales

L'ACP peut être vue comme un outil de visualisation de données multidimensionnelles : il est facile de visualiser un nuage de points en dimension 2, mais pas en dimension supérieur à trois. Si l'on dispose d'une seule variable, il est facile de représenter les n points (n observations/échantillons biologiques/individus statistiques) en une dimension (on place simplement les points sur une droite). Si l'on ne mesure deux variables uniquement, on peut représenter les n points (n échantillons biologiques) dans le plan. La visualisation du nuage de points dans un espace de dimension $d \geq 3$ est difficile.
Cependant, l'information apportée par de nombreuses variables est parfois redondante ou superflue. On va alors remplacer les variables initiales par un nombre réduit de variables "construites" qui résument le mieux possible l'information contenue dans le jeu de données. Ces nouvelles variables seront des combinaisons linéaires des variables initiales et seront non corrélées les unes aux autres. Pour trouver ces combinaisons linéaires de variables "optimales", on va utiliser la décomposition spectrale de la matrice de variance covariance empirique.
On note X la matrice des données, pour n individus (observations, échantillons biologiques) et p variables :

$$X_{(n,p)} = \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,p} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,p} \end{pmatrix}$$

On note X_i la $i$ème ligne du tableau (vecteur ligne) et X^j (vecteur colonne). On souhaite trouver des nouvelles variables, notées $C^1, \ldots, C^p$ telles que C^j soit une combinaison linéaire des variables $X^1, \ldots, X^p$ et telles que les nouvelles variables C^1 et C^2 résument à elles seules le mieux possible l'information du jeu de données (i.e. telles que la variance des nouvelles variables C^1 et C^2 soit les plus grandes possibles, avec $V(C^1) > V(C^2)$).

Etape 1 : on centre et on réduit le nuage de points (la variance de chaque variable réduite vaut 1, cette étape de réduction est cependant facultative, mais les formules ci-dessous ne seront pas forcément vraies).

Etape 2 : on calcule la matrice de variance covariance empirique : $S = \frac{1}{n} X^T X$.

Etape 3 : on effectue la décomposition spectrale de la matrice S.

Etape 4 : **Coordonnées des individus sur les axes principaux.** Les deux premiers axes principaux sont les axes portés par les vecteurs propres $(\overrightarrow{u_1}, \overrightarrow{u_2})$ associés aux deux plus grandes valeurs propres λ_1 et λ_2. On calcule les coordonnées des points dans le plan défini par les deux axes principaux. Le vecteur $C^1 = X\overrightarrow{u_1}$ donne les coordonnées des points en abscisse et le vecteur $C^2 = X\overrightarrow{u_2}$ donne les coordonnées des points en ordonnée.
On note

$$C_{(n,2)} = \begin{pmatrix} c_{1,1} & c_{1,2} \\ c_{2,1} & c_{2,2} \\ \vdots & \vdots \\ c_{n,1} & c_{n,2} \end{pmatrix}$$

Etape 5 : **Variance projetée sur les axes principaux.** La variance empirique du nuage de points est la somme des variances de chaque variable. C'est la somme des coefficients de la diagonale de la matrice S. C'est aussi la somme des valeurs de propres de S. La variance des points projetés dans le plan porté par les vecteurs propres $(\overrightarrow{u_1}, \overrightarrow{u_2})$ est $\lambda_1 + \lambda_2$.On peut calculer le pourcentage de la variance expliquée par l'axe 1 et 2 de l'ACP en calculant :

$$\frac{\lambda_1 + \lambda_2}{\sum_{k=1}^{p} \lambda_k} \times 100$$

Etape 6 : **Contribution relative d'un axe à un individu.** La variance totale du nuage de points est $\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{p}(x_{i,j})^2$. La variance apporté par l'individu i est $\frac{1}{n}\sum_{j=1}^{p}(x_{i,j})^2$. La variance projetée sur l'axe 1 est $\frac{1}{n}(c_{i,1})^2$. La contribution relative de l'axe 1 à l'individu i est le ratio $\frac{(c_{i,1})^2}{\sum_{j=1}^{p}(x_{i,j})^2}$. Cette quantité représente le carré du cosinus de l'angle formé par l'individu i et le vecteur $\overrightarrow{u_1}$. Cette quantité

est souvent nommé `cos2` dans les paramètres des sorties des fonctions `PCA` (package `FactoMineR`) ou `dudi.pca` (package `ade4`).

Etape 7 : **Contribution relative d'un individu à un axe.** De même, on peut décomposer la variance de la variable C^1 en utilisant $\lambda_1 = \frac{1}{n}\sum_{i=1}^{n}(c_{i,1})^2$. La contribution relative de l'individu i à l'axe 1 est $\frac{1}{n}\frac{(c_{i,1})^2}{\lambda_1}$.

Etape 8 : **Corrélation entre les anciennes variables et les nouvelles variables.** On calcule la corrélation entre l'ancienne variable j et la composante principale 1.

$$\text{cor}(X^j, C^1) = \text{cor}(X^j, X\overrightarrow{u_1}) = \frac{\text{cov}(X^j, X\overrightarrow{u_1})}{\sqrt{V(X^j)}\sqrt{V(Xu_1)}}$$

On a $V(X^j) = 1$ car les données ont été réduites. On sait aussi que la variance de $C^1 = X\overrightarrow{u_1}$ vaut λ_1. On considère $X^j = X\overrightarrow{e_j}$ où $\overrightarrow{e_j}$ est le vecteur dont la $j$ème coordonnées vaut 1 et 0 sinon.

$$\text{cov}(X^j, X\overrightarrow{u_1}) = \text{cov}(X\overrightarrow{e_j}, X\overrightarrow{u_1}) = \frac{1}{n}X\overrightarrow{e_j}^T X^T X\overrightarrow{u_1} = \lambda_1 \overrightarrow{e_j}^T\overrightarrow{u_1}$$

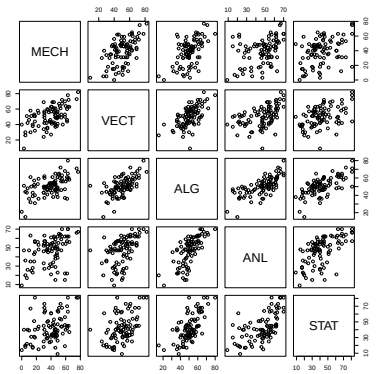
car $\overrightarrow{u_1}$ est vecteur propre de la matrice $S = \frac{1}{n}X^T X$. On obtient :

$$\text{cor}(X^j, C^1) = \sqrt{\lambda_1}\, u_{1,j}$$

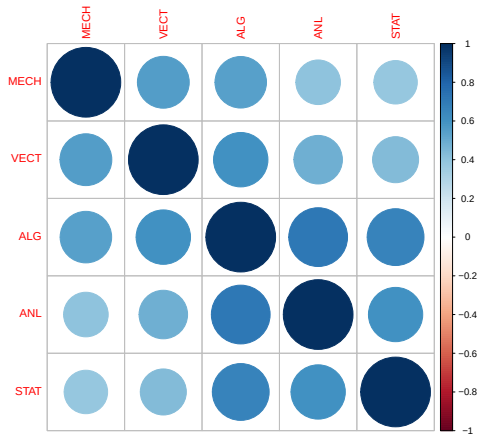
où $u_{1,j}$ est la $j$ème coordonnées du vecteur $\overrightarrow{u_1}$.

Exemple sur un jeu de données

```
## Un exemple sur des données réelles
library(bnlearn) ## pour avoir les données Marks
# Marks data from Mardia, 1974.
data(marks)
str(marks)

## 'data.frame': 88 obs. of  5 variables:
##  $ MECH: num  77 63 75 55 63 53 51 59 62 64 ...
##  $ VECT: num  82 78 73 72 63 61 67 70 60 72 ...
##  $ ALG : num  67 80 71 63 65 72 65 68 58 60 ...
##  $ ANL : num  67 70 66 70 70 64 65 62 62 62 ...
##  $ STAT: num  81 81 81 68 63 73 68 56 70 45 ...

dim(marks)

## [1] 88  5

plot(marks)
```

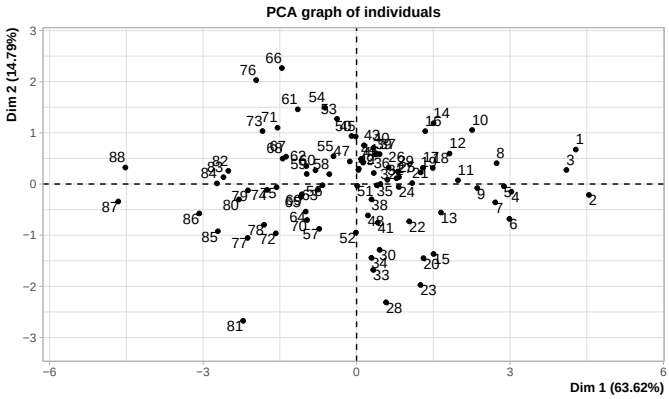


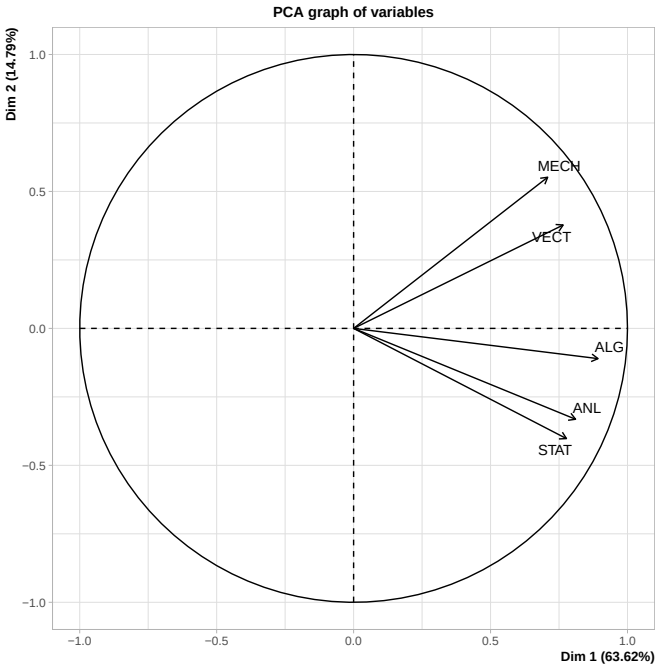
```r
## analyse des correlation
library(corrplot)

## corrplot 0.92 loaded

corrplot(cor(marks))
```



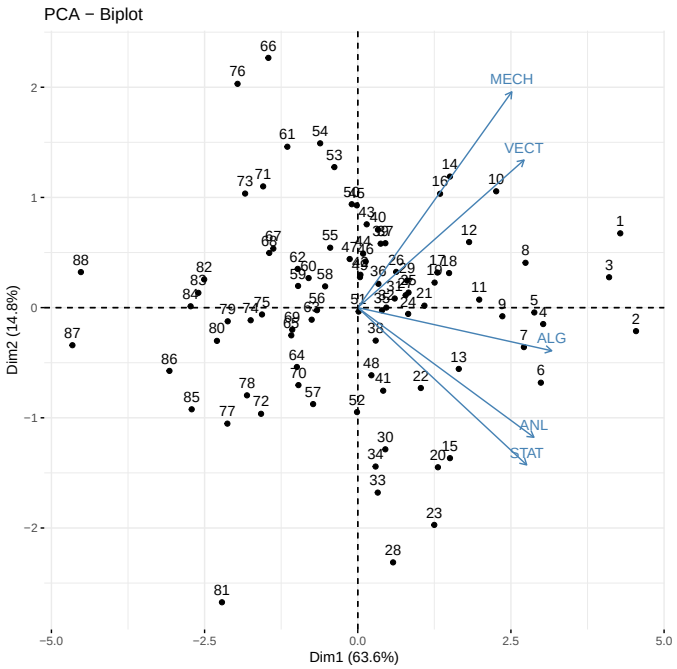
ACP à l'aide de la fonction `PCA` du package `FactoMineR`

```r
## on fait la PCA avec FactoMineR
library(FactoMineR)
## etape 0 : on centre et on réduit les données
X <- scale(marks,center = TRUE,scale = TRUE)
colMeans(X)

##          MECH          VECT           ALG           ANL          STAT
##  8.015211e-17 -1.945256e-16  5.653622e-17  1.252943e-16  1.162265e-16

apply(X,2,var)

## MECH VECT  ALG  ANL STAT
##    1    1    1    1    1

## on applique la fonction PCA
resPCA <- PCA(X,scale.unit = FALSE)
```





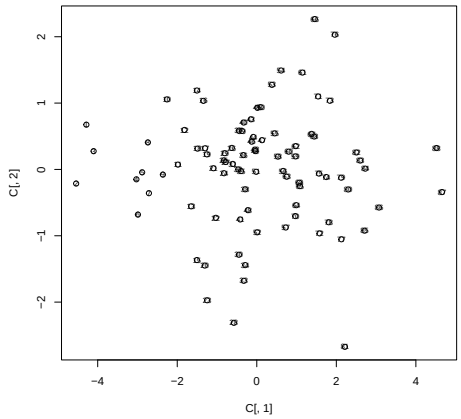

```
## on utilise le package factoextra
## pour avoir une représentation visuelle
library(factoextra)

## Loading required package: ggplot2
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa

fviz_pca_biplot(resPCA)
```



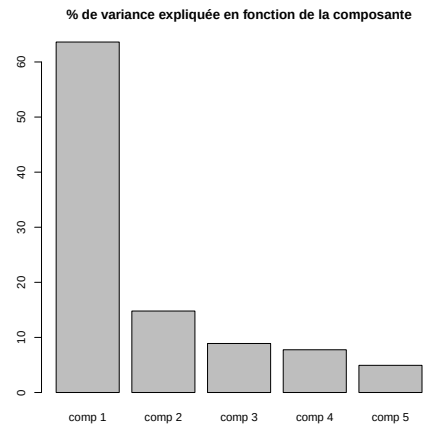
ACP à l'aide de la fonction `eigen`

```
library(bnlearn)
## Etape 0 : on centre et réduit les données
X <- scale(marks,center = TRUE,scale = TRUE)
colMeans(X)

##          MECH          VECT           ALG           ANL          STAT
##  8.015211e-17 -1.945256e-16  5.653622e-17  1.252943e-16  1.162265e-16

apply(X,2,var)

## MECH VECT  ALG  ANL STAT
##    1    1    1    1    1

## Etape 1 : on calcule la matrice de variance covariance empirique
n=dim(X)[1]
S=(n-1)/n*cov(X) # ou S=1/n*t(X)%*%X
## Etape 2: décomposition spectrale de S
eigen(S)

## eigen() decomposition
## $values
## [1] 3.1448326 0.7311676 0.4399087 0.3834845 0.2437883
##
## $vectors
##            [,1]       [,2]        [,3]       [,4]       [,5]
## [1,] -0.3996045  0.6454583 -0.62078249 -0.1457865 -0.1306722
## [2,] -0.4314191  0.4415053  0.70500628  0.2981351 -0.1817479
## [3,] -0.5032816 -0.1290675  0.03704901 -0.1085987  0.8466894
## [4,] -0.4569938 -0.3879057  0.13618182 -0.6662561 -0.4221885
## [5,] -0.4382444 -0.4704545 -0.31253342  0.6589164 -0.2340223

## Etape 3: calcul des coordonnées des individus
```

```r
C=X%*%eigen(S)$vectors[,1:2]
par(mfrow=c(1,1))
plot(C[,1],C[,2],lwd=0.001)
text(C[,1],C[,2], labels=1:n, cex= 0.7)
```




```r
## vérification de la cohérence
## avec la sortie FactoMineR
resPCA$ind$coord[10,1:2]

##    Dim.1    Dim.2
## 2.260005 1.055602

C[10,] ## meme valeur au signe près

## [1] -2.260005  1.055602

## Etape 4:Variance projetée sur les axes principaux.
## liste des valeurs propres de S
eigen(S)$values

## [1] 3.1448326 0.7311676 0.4399087 0.3834845 0.2437883

## % expliqué par chaque axe
eigen(S)$values[1:2]/(sum(eigen(S)$values))*100

## [1] 63.61960 14.79144

## verif avec FactoMineR
resPCA$eig

##        eigenvalue percentage of variance cumulative percentage of variance
## comp 1  3.1448326             63.619603                          63.61960
## comp 2  0.7311676             14.791437                          78.41104
## comp 3  0.4399087              8.899303                          87.31034
## comp 4  0.3834845              7.757848                          95.06819
## comp 5  0.2437883              4.931810                         100.00000

## représentation graphique utile pour savoir combien
## d'axes regarder
barplot(resPCA$eig[,2],main="% de variance expliquée en fonction de la composante")
```




```r
## Etape 5: contribution relative d'un axe à un individu
1/n*sum(X^2) ## variance totale (VT) du nuage de point
```

```
## [1] 4.943182
```

```r
sum(diag(S)) ## autre manière de calculer cette VT
```

```
## [1] 4.943182
```

```r
sum(eigen(S)$values) ## autre manière
```

```
## [1] 4.943182
```

```r
## part de l'individu 10 dans la VT
1/n*sum(X[10,]^2)
```

```
## [1] 0.0747132
```

```r
## ratio avec la VT
1/n*sum(X[10,]^2)/(1/n*sum(X^2))
```

```
## [1] 0.01511439
```

```r
## part de la variance apportée par 10
## et conservée par l'axe 1 et 2
1/n*C[10,]^2
```

```
## [1] 0.05804119 0.01266246
```

```r
## ratio avec la variance totale apportée par 10
1/n*C[10,]^2/((1/n*sum(X[10,]^2)) )
```

```
## [1] 0.7768532 0.1694809
```

```r
C[10,]^2/sum(X[10,]^2)
```

```
## [1] 0.7768532 0.1694809
```

```r
## cette quantité correspond aussi au carré du cosinus
## de l'angle formé par l'individu 10 avec le vecteur u_1
## si c'est proche de 1, l'individu est bien représenté par l'axe 1
## si proche de 0 , mal représenté

## verif avec FactoMineR
resPCA$ind$cos2[10,]
```

```
##      Dim.1      Dim.2      Dim.3      Dim.4      Dim.5
## 0.77685316 0.16948089 0.02236127 0.02482602 0.00647866

## Etape 6 : contribution relative d'un individu à un axe
## la variance de la variable C^1 est lambda1
var(C[,1])*(n-1)/n

## [1] 3.144833

eigen(S)$values

## [1] 3.1448326 0.7311676 0.4399087 0.3834845 0.2437883

1/n*sum(C[,1]^2)

## [1] 3.144833

## la contribution de l'individu 10 à cette variance est
C[10,1]^2/n

## [1] 0.05804119

## le ration avec la variance de C1 est
C[10,1]^2/n/eigen(S)$values

## [1] 0.01845605 0.07938151 0.13193916 0.15135210 0.23808024

C[10,1]^2/n/(1/n*sum(C[,1]^2))

## [1] 0.01845605

C[10,1]^2/sum(C[,1]^2)

## [1] 0.01845605

## ici l'individu i contribue à expliquer
## 1,84% de la variabilité de l'axe 1

## cohérence avec les sorties de FactoMineR
resPCA$ind$contrib[10,]

##     Dim.1     Dim.2     Dim.3     Dim.4     Dim.5
## 1.8456049 1.7318135 0.3797793 0.4836783 0.1985499

sum(resPCA$ind$contrib[,1])

## [1] 100

## Etape 7: Cercle de corrélations des variables
## On regarde la corrélation entre les nouvelles variables C^1 et C^2
## et les anciennes variables

## cor(X^3;C^1)
sqrt(eigen(S)$values)[1]*eigen(S)$vectors[3,1]

## [1] -0.8925032
```

```
## coéhrence avec la sortie FactoMineR
resPCA$var$coord[3,1] # au signe près

## [1] 0.8925032

## fonction dimdesc de FactoMineR
dimdesc(resPCA)

## $Dim.1
##
## Link between the variable and the continuous variables (R-square)
## ================================================================================
##      correlation      p.value
## ALG    0.8976179 2.408585e-32
## ANL    0.8150623 4.307358e-22
## STAT   0.7816221 2.560867e-19
## VECT   0.7694490 1.994936e-18
## MECH   0.7127067 6.795513e-15
##
## $Dim.2
##
## Link between the variable and the continuous variables (R-square)
## ================================================================================
##      correlation      p.value
## MECH   0.5550836 1.995212e-08
## VECT   0.3796873 2.637941e-04
## ANL   -0.3335926 1.492515e-03
## STAT  -0.4045832 9.244911e-05
##
## $Dim.3
##
## Link between the variable and the continuous variables (R-square)
## ================================================================================
##      correlation      p.value
## MECH   0.4140973 6.055635e-05
## VECT  -0.4702794 3.778685e-06
```

Pour aller plus loin dans la compréhension de l'ACP

http://www2.agroparistech.fr/IMG/pdf/AnalyseComposantesPrincipales-AgroParisTech.pdf