Chapitre III : Réduction de dimension

Apprentissage en grande dimension

M. Champion

Université de Paris

2020-2021

Introduction

Grande dimension

On considère une matrice de données X et un vecteur d'observations Y à expliquer. Les observations portent sur p variables, mesurées sur n individus. Il existe plusieurs situations de **grande dimension** :

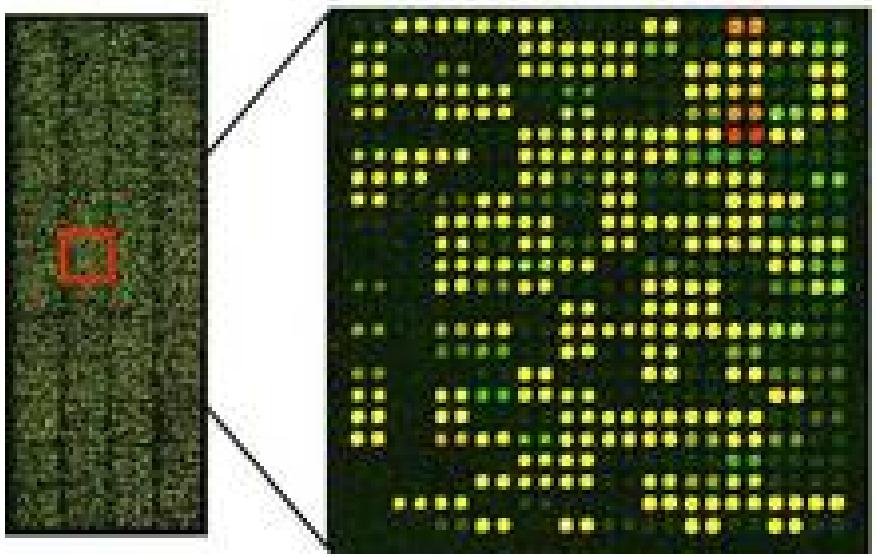
- n grand et p de taille raisonnable
  - situation favorable d'un point de vue théorique,
  - problème de stockage informatique,
  - solutions informatiques de BigData (Hadoop,...).

Introduction

Grande dimension

On considère une matrice de données X et un vecteur d'observations Y à expliquer. Les observations portent sur p variables, mesurées sur n individus. Il existe plusieurs situations de **grande dimension** :

- n de taille raisonnable et p grand ($p >> n$)
  - problèmes théoriques sous-jacents,
  - réduction de dimension, sélection de variables.



Introduction

Grande dimension

On considère une matrice de données X et un vecteur d'observations Y à expliquer. Les observations portent sur p variables, mesurées sur n individus. Il existe plusieurs situations de **grande dimension** :

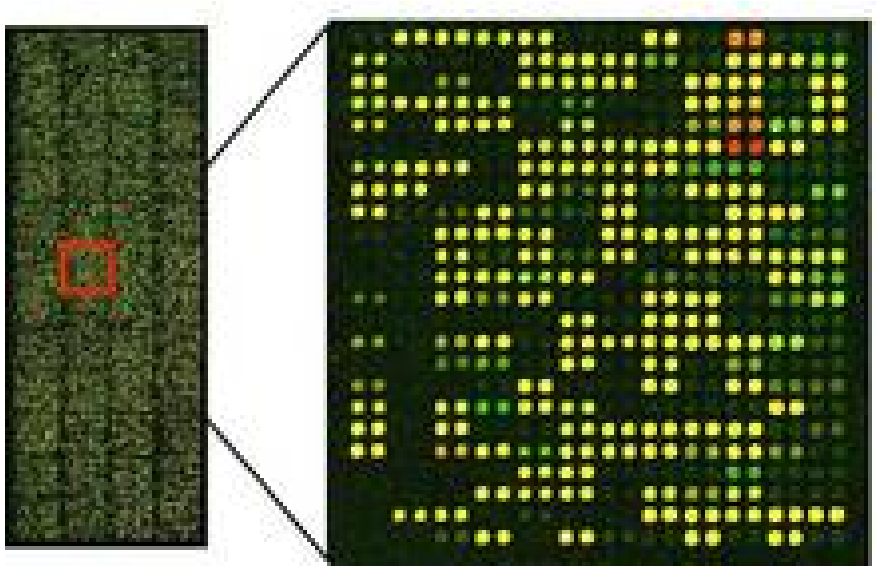
- n et p grands
  - situation la plus compliquée,
  - outils informatiques de BigData + outils stats de grande dimension.

Introduction

Données d'expression des gènes

Les données *fil rouge* de ce cours seront des données d'apprentissage supervisé en génomique :

- $X \in \mathcal{M}(n, p)$ désigne la mesure de l'**expression** de p gènes chez n individus.
- $y \in \mathbb{R}^n$ est une variable **phénotypique** mesurée chez tous les individus, par exemple un indice de virulence de la tumeur.



Introduction

Contourner l'écueil de la grande dimension

En présence de nombreuses variables explicatives, on suppose généralement que peu d'entre elles sont pertinentes pour modéliser/prédire Y. Il existe quatre familles de méthodes permettant de contourner le fléau de grande dimension :

- **tests multiples** : utilisés en pré-traitement pour filtrer les variables,
- **réduction de dimension** : utilisés en pré-traitement pour réduire la dimension de l'espace des variables,
- **choix de modèles** : pour choisir le meilleur sous-modèle,
- **régressions sous contraintes** (ou pénalisées) : pour contraindre le nombre de paramètres dans le modèle.

III. 1. Rappels sur le modèle linéaire

Rappels sur le modèle linéaire

Modèle linéaire gaussien

On appelle **modèle linéaire gaussien** un modèle statistique qui peut s'écrire sous la forme :

$$\forall i \in [\![1, n]\!], \quad Y_i = \beta_0 + \beta_1 X_i^1 + ... + \beta_p X_i^p + \varepsilon_i,$$

où les $(\varepsilon_i)_{1 \leq i \leq n}$ sont des termes d'erreur non observés, i.i.d :

$$\varepsilon_i \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma^2).$$

Le modèle se réécrit matriciellement :

$$Y = X\beta + \varepsilon,$$

où X est une matrice de taille $n \times (p+1)$ de terme général X_i^j, excepté la première colonne qui contient uniquement des 1, $Y = {}^t(Y_1, ..., Y_n)$, $\varepsilon = {}^t(\varepsilon_1, ..., \varepsilon_n)$ et $\beta = {}^t(\beta_0, ..., \beta_p)$.

Rappels sur le modèle linéaire

Estimation des paramètres

Estimation de :

- β par moindres carrés :
$$\hat{\beta} = ({}^tXX)^{-1}{}^tXY.$$
- la variance du bruit σ^2 :
$$\hat{\sigma}^2 = \frac{1}{n-(p+1)} \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2.$$

Parmi les estimateurs de β, $\hat{\beta}$ est le meilleur estimateur sans biais au sens du coût quadratique.

Rappels sur le modèle linéaire

Qualité d'ajustement

La qualité d'ajustement du modèle est mesurée par le coefficient de détermination, défini par :

$$R^2 = \frac{SCM}{SCT} = 1 - \frac{SCR}{SCT} = 1 - \frac{\sum_i (Y_i - \hat{Y}_i)^2}{\sum_i (Y_i - \bar{Y})^2}.$$

Il détermine à quel point l'équation de régression est adaptée pour décrire la distribution des points.

Remarque : En règle général, on utilise le R^2 ajusté :

$$R^2_{\text{ajusté}} = 1 - \frac{SCR\,/(n-p-1)}{SCT\,/(n-1)}.$$

Pour rappel :

$$SCT = \sum_{i=1}^{n} (Y_i - \bar{Y})^2 = \sum_{i=1}^{n} (\widehat{Y}_i - \bar{Y})^2 + \sum_{i=1}^{n} (Y_i - \widehat{Y}_i)^2 = SCM + SCR\,.$$

Rappels sur le modèle linéaire

Test de la contribution globale

Il s'agit d'une généralisation du test de Fisher à p variables explicatives :

- compare les modèles M_1 (constant) et M_{p+1} (complet)

$$M_1 \quad : \quad \forall i \in [\![1, n]\!], \quad Y_i = \beta_0 + \varepsilon_i, \quad \varepsilon_i \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma^2)$$

$$M_{p+1} \quad : \quad \forall i \in [\![1, n]\!], \quad Y_i = \beta_0 + \sum_{j=1}^{p} \beta_j X_i^j + \varepsilon_i, \quad \varepsilon_i \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma^2).$$

- revient à tester, au risque δ fixé, l'hypothèse nulle

$$(H_0) : \text{ modèle } M_1 \text{ (aucune variable n'a d'influence)}$$

contre l'alternative

$$(H_1) : \text{ modèle } M_{p+1} \text{ (au moins une des variables a de l'influence)}$$

Rappels sur le modèle linéaire

Test de la contribution globale

La **statistique de test** est

$$T_n = \frac{\text{SCM}/p}{\text{SCR}/(n-p-1)} \sim_{H_0} F(p, n-p-1),$$

ou encore :

$$T_n = \frac{(\text{SCR}_{\text{M}_1} - \text{SCR})/p}{\text{SCR}/(n-p-1)} \sim_{H_0} F(p, n-p-1),$$

où SCR_{M_1} est la Somme des Carrés Résiduels pour le modèle M_1.

Rappels sur le modèle linéaire

Test de la contribution globale

Notons c_δ le quantile d'ordre δ de $F(p, n-p-1)$ tel que :

$$\mathbb{P}(T_n \leq c_\delta) = 1 - \delta$$

La **règle de décision** est alors la suivante :

- si $T_n > c_\delta$, on rejette H_0,
- si $T_n \leq c_\delta$, on ne rejette pas H_0.

En pratique, on prend un échantillon de taille n, on calcule la réalisation t_n de T_n sur cet échantillon et on compare sa valeur à c_δ, qui est lu sur la table de Fisher. On peut aussi calculer la p-valeur$= \mathbb{P}_{H_0}(T_n > t_n)$ que l'on compare au risque δ. Si la p-valeur est inférieure à δ, le test est significatif.

Rappels sur le modèle linéaire

Test du modèle réduit

On teste si un ensemble de q variables explicatives ne suffit pas à expliquer Y :

- **Hypothèses** :

$$(\mathrm{H}_0) : \text{modèle réduit}$$

$$\text{Modèle } M_{q+1} : \forall i \in [\![1, n]\!], \quad Y_i = \beta_0 + \sum_{j=1}^{q} \beta_j X_i^j + \varepsilon_i, \quad \varepsilon_i \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma^2),$$

contre

$$(\mathrm{H}_1) : \text{modèle complet}$$

$$\text{Modèle } M_{p+1} : \forall i \in [\![1, n]\!], \quad Y_i = \beta_0 + \sum_{j=1}^{p} \beta_j X_i^j + \varepsilon_i, \quad \varepsilon_i \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma^2).$$

Ou encore :

$$(H_0) : \forall j = q+1, ..., p, \quad \beta_j = 0 \text{ contre } (H_1) : \exists j \in \{q+1, ..., p\}, \quad \beta_j \neq 0.$$

Rappels sur le modèle linéaire

Test du modèle réduit

Théorème

Sous (H_0), *on a le résultat suivant :*

$$\frac{\text{SCR}_{M_{q+1}}}{\sigma^2} \sim_{H_0} \chi^2(n-q-1).$$

De plus, $\text{SCR}_{M_{q+1}} - \text{SCR}$ *et* SCR *sont indépendants et*

$$\frac{\text{SCR}_{M_{q+1}} - \text{SCR}}{\sigma^2} \sim_{H_0} \chi^2(p-q).$$

La **statistique de test** est alors définie par

$$T_n = \frac{(\text{SCR}_{M_{q+1}} - \text{SCR})/(p-q)}{\text{SCR}\,/(n-p-1)} \sim_{H_0} F(p-q, n-p-1).$$

Rappels sur le modèle linéaire

Test du modèle réduit

Notons c_δ le quantile d'ordre δ de $F(p-q, n-p-1)$ tel que :

$$\mathbb{P}(T_n \leq c_\delta) = 1 - \delta$$

La **règle de décision** est alors la suivante :

- si $T_n > c_\delta$, on rejette H_0,
- si $T_n \leq c_\delta$, on ne rejette pas H_0.

En pratique, on mesure T_n sur un échantillon de taille n. Si $c_n > t_\delta$, on conserve le modèle complet, on considère que le passage de M_{q+1} à M_{p+1} est significatif : au moins l'une des variables $X^{q+1}, ..., X^p$ a une influence significative sur Y (en plus de $X^1, ..., X^q$). Si $t_n \leq c_\delta$, on ne rejette pas (H_0), on considère que le passage de M_{q+1} à M_{p+1} n'est pas significatif : l'influence des variables explicatives $X^{q+1}, ..., X^p$ n'est pas significative pour expliquer Y.

Rappels sur le modèle linéaire

Choix de modèles

Si on hésite entre plusieurs modèles $(M_{q+1})_{q<p}$, on peut aussi comparer les critères :

- R^2 ajusté :

$$\max_{q\leq p} \; R^2_{\text{ajusté}}(q) = \max_{q\leq p} \; 1 - \frac{\text{SCR}_{M_{q+1}}/(n-q-1)}{\text{SCT}/(n-1)}.$$

- C_p de Mallows :

$$\min_{q\leq p} C_p(q) = \min_{q\leq p} \; \frac{\text{SCR}_{M_{q+1}}}{\hat{\sigma}^2} - n + 2(q+1).$$

- Critère AIC :

$$\min_{q\leq p} \text{AIC}(q) = \min_{q\leq p} \; n \ln \frac{\text{SCR}_{M_{q+1}}}{n} + 2(q+1).$$

- Critère BIC :

$$\min_{q\leq p} \text{BIC}(q) = \min_{q\leq p} \; n \ln \frac{\text{SCR}_{M_{q+1}}}{n} + (q+1) \ln n.$$

Rappels sur le modèle linéaire

Vers la grande dimension

En grande dimension :

- le modèle linéaire est mal défini : il existe une infinité de β conduisant au même vecteur $X\beta$,
- les critères énoncés précédemment ne sont plus utilisables car ils sont basés sur les résidus $\hat{Y} - Y$, qui valent 0 lorsque $p \geq n$.

Pour contourner le problème, on fait en général une hypothèse de **parcimonie** : seuls p^* coefficients β_j sont non nuls avec $p^* << n$. La difficulté réside dans le fait que l'on ne connaît pas p^*.

Rappels sur le modèle linéaire

Modèle linéaire généralisé

On appelle **modèle linéaire généralisé** un modèle statistique qui peut s'écrire sous la forme :

$$\begin{cases} Y \sim f_\theta, \quad f_\theta \in \mathcal{F}(\theta), \\ \mathbb{E}(Y) = \mu, \\ g(\mu) = X\beta = \beta_0 + \beta_1 X1 + ... + \beta_p X^p. \end{cases}$$

Pour que ce modèle soit bien défini, il faut choisir :

1. la famille paramétrique $\mathcal{F}(\Theta)$ à laquelle appartient la loi de Y : ici, il s'agit de la famille **exponentielle** (lois normales, exponentielles, gamma, Bernoulli, binomiales, Poisson),
2. la **fonction de lien** g qui relie $\mathbb{E}(Y)$ et $(X^1, ...X^p)$.

Rappels sur le modèle linéaire

Régression logistique

Si on choisit la loi Bernoulli en supposant que $Y \sim \mathcal{B}(p)$ et la fonction de lien **logistique** $g(x) = \log \frac{x}{1-x}$, on obtient un **modèle de régression logistique** :

$$\begin{cases} Y \sim \mathcal{B}(p), \\ \log\left(\frac{\mathbb{P}(Y=1)}{\mathbb{P}(Y=0)}\right) = X\beta. \end{cases}$$

Remarques :

- le modèle de régression logistique ne s'écrit pas $Y = X\beta + \varepsilon$, en tout cas pas avec une erreur de loi connue,
- le modèle de régression logistique est utilisé lorsque Y est qualitative (variable clinique 0/1 par exemple).

Rappels sur le modèle linéaire

Régression logistique

- **Estimation** : l'estimateur du maximum de vraisemblance a de bonnes propriétés théoriques (consistance, normalité asymptotique, optimalité) mais il n'en existe pas toujours de formulation explicité, il faut donc recourir à des algorithmes d'optimisation (Newton-Raphson par exemple) pour l'approcher numériquement.

- **Choix de modèle** : comparaisons via un critère de vraisemblance pénalisée ou la déviance :

$$D = -2\left(\ell(Y, \hat{\beta}) - \ell_{sat}(Y)\right) \geq 0,$$

où le modèle saturé désigne le modèle possédant autant de paramètres que d'observations et estimant donc exactement les données. Le modèle est d'autant meilleur que la déviance est petite.

Rappels sur le modèle linéaire

Applications sur R - modèle linéaire gaussien

```
library(glmnet)
library(lasso2) # téléchargement des données
data("Prostate")
str(Prostate)
```

```
## 'data.frame':    97 obs. of  9 variables:
##  $ lcavol : num  -0.58 -0.994 -0.511 -1.204 0.751 ...
##  $ lweight: num  2.77 3.32 2.69 3.28 3.43 ...
##  $ age    : num  50 58 74 58 62 50 64 58 47 63 ...
##  $ lbph   : num  -1.39 -1.39 -1.39 -1.39 -1.39 ...
##  $ svi    : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ lcp    : num  -1.39 -1.39 -1.39 -1.39 -1.39 ...
##  $ gleason: num  6 6 7 6 6 6 6 6 6 6 ...
##  $ pgg45  : num  0 0 20 0 0 0 0 0 0 0 ...
##  $ lpsa   : num  -0.431 -0.163 -0.163 -0.163 0.372 ...
```

Rappels sur le modèle linéaire

Applications sur R - modèle linéaire gaussien

```
model <- lm(lcavol~.,data=Prostate)
summary(model)
```

```
##
## Call:
## lm(formula = lcavol ~ ., data = Prostate)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.88603 -0.47346 -0.03987  0.55719  1.86870
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.260101   1.259683  -1.794   0.0762 .
## lweight     -0.073166   0.174450  -0.419   0.6759
## age          0.022736   0.010964   2.074   0.0410 *
## lbph        -0.087449   0.058084  -1.506   0.1358
## svi         -0.153591   0.253932  -0.605   0.5468
```

Rappels sur le modèle linéaire

Applications sur R - modèle logistique

```
library(glmnet)
library(lsplsGlm) # téléchargement des données
data("BreastCancer")
X <- BreastCancer$X # données d'expression
Y <- BreastCancer$Y # présence/absence de métastase
dim(X)
```

```
## [1]   78 4348
```

```
table(Y)
```

```
## Y
##  0  1
## 44 34
```

Rappels sur le modèle linéaire

Applications sur R - modèle logistique

```
model_logit <- glm(Y ~ X[,1:10], family = "binomial")
summary(model_logit)
```

```
##
## Call:
## glm(formula = Y ~ X[, 1:10], family = "binomial")
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8657  -0.9549  -0.6054   1.0107   1.9990
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.4316     0.4685  -0.921   0.3569
## X[, 1:10]1    0.6438     0.6253   1.030   0.3032
## X[, 1:10]2   -0.8452     0.7567  -1.117   0.2640
## X[, 1:10]3   -0.3906     1.1766  -0.332   0.7399
## X[, 1:10]4   -1.1177     1.2882  -0.868   0.3856
```

III. 2. Sélection par tests multiples

Sélection par tests multiples

Introduction

- Une manière de sélectionner les variables pour en réduire le nombre consiste au préalable (*pré-traitement*) à effectuer des **tests statistiques** :
  - en génomique : on teste si l'expression d'un gène est différente entre deux conditions expériementales.

- on se ramène alors à des statistiques en petite dimension dans l'espace réduit à ces variables,

- on s'attend par contre à détecter de nombreux faux positifs puisque chaque test est associé à un risque de première espèce α, qui s'accumulent.

Sélection par tests multiples

Introduction

https://xkcd.com/882/

Sélection par tests multiples

Notations et problématique

- On suppose qu'on a m tests à effectuer (avec $m < p$).
- Chaque test est associé à une hypothèse nulle $(H_{0,i})$ et alternative $(H_{1,i})$, par exemple :

$$(H_{0,i}) \quad : \quad \mu_i = 0 \quad \text{contre} \quad (H_{1,i}) \quad : \quad \mu_i \neq 0.$$

  - en génomique : μ_i représente la différence d'expression d'un gène entre deux conditions expérimentales.
- A chaque test est associé une p-valeur $p_{(i)}$ correspondante.

Règle de décision :

- A partir de quel seuil τ (pour les $p_{(i)}$) rejette-t-on $(H_{0,i})$?
- Si on ordonne les p_i par ordre croissant, à partir de quel rang k considère-t-on que les p-valeurs $p_1, ..., p_k$ conduisent à rejeter $(H_{0,i})$?

Sélection par tests multiples

Notations et problématique

Table de vérité :

	Décision	
Réalité	(H_0)	(H_1)
(H_0)	TN	FP
(H_1)	FN	TP

Objectif : maximiser le nombre de TP tout en minimisant les FP.

On note I_0 l'ensemble des indices i pour lesquels $(H_{0,i})$ est vrai. Si on fixe τ un seuil indépendant de m (typiquement 5%) au-delà duquel on rejette (H_0) :

$$\mathbb{E}(\text{FP}) = \mathbb{E}\left(\sum_{i=1}^{m} 1_{\{i \in I_0, p_{(i)} < \tau\}}\right) = \sum_{i \in I_0} \mathbb{P}(p_{(i)} < \tau) = \#I_0 \tau \quad >> 0.$$

Sélection par tests multiples

Contrôles pour tests multiples

Contrôler le risque de première espèce α n'est pas suffisant mais on peut décider de contrôler le :

- **Family Wise Error Rate** ou taux d'erreur par famille :

$$\text{FWER} = \mathbb{P}(\text{FP} \geq 1),$$

- **False Discovery Rate** ou taux de faux positifs :

$$\text{FDR} = \mathbb{E}\left(\frac{\text{FP}}{\text{FP} + \text{TP}}\right).$$

Proposition

Si $m = 1$ *(un seul test effectué),* $\text{FWER} = \text{FDR}$ *correspond au risque usuel de 1ière espèce. Si* $m > 1$,

$$\text{FDR} \leq \text{FWER}.$$

Sélection par tests multiples

Contrôles pour tests multiples

Contrôler le risque de première espèce α n'est pas suffisant mais on peut décider de contrôler le :

- **Family Wise Error Rate** ou taux d'erreur par famille :

$$\text{FWER} = \mathbb{P}(\text{FP} \geq 1),$$

- **False Discovery Rate** ou taux de faux positifs :

$$\text{FDR} = \mathbb{E}\left(\frac{\text{FP}}{\text{FP} + \text{TP}}\right).$$

Le contrôle du FWER est plus fort puisqu'il permet de s'assurer qu'avec grande probabilité, aucun faux positif n'est sélectionné. Le contrôle du FDR est plus souple, il permet de moins rejeter de positifs, au prix d'un plus fort taux de faux positifs.

Sélection par tests multiples

Procédure de Bonferroni pour contrôle du FWER

Procédure de Bonferroni : les m tests individuels sont effectués en remplaçant α par α/m.

Proposition

La procédure de Bonferroni assure

$$\text{FWER} \leq \alpha.$$

- La procédure garantit un contrôle du FWER quelque soit la dépendance entre les p-valeurs des m tests.
- La procédure est par contre trop conservative si m est grand (trop peu de positifs détectés).

Sélection par tests multiples

Procédure de Holm-Bonferroni pour contrôle du FWER

1. Effectuer les m tests et ordonner les m p-valeurs obtenues

$$p_{(1)} \leq p_{(2)} \leq \ldots \leq p_{(m)}.$$

2. Déterminer

$$I = \max \left\{ k, \quad \forall i \leq k, \quad p_{(i)} \leq \frac{\alpha}{m - i + 1} \right\}.$$

3. Rejeter les p-valeurs inférieures à $\frac{\alpha}{m-I+1}$.

Proposition

La procédure de Holmes-Bonferroni assure

$$\text{FWER} \leq \alpha,$$

et est toujours supérieure à la procédure de Bonferroni.

Sélection par tests multiples

Procédure de Benjamini-Hochberg pour contrôle du FDR

Procédure la plus utilisée en pratique :

1. Effectuer les m tests et ordonner les m p-valeurs obtenues

$$p_{(1)} \leq p_{(2)} \leq \ldots \leq p_{(m)}.$$

2. Déterminer

$$I = \max \left\{ k, \quad \forall i \leq k, \quad p_{(i)} \leq \alpha \frac{i}{m} \right\}.$$

3. Rejeter les p-valeurs inférieures à $\alpha \frac{I}{m}$.

Proposition

La procédure de Benjamini-Hochberg assure

$$\text{FDR} \leq \alpha.$$

Sélection par tests multiples

p-valeurs ajustées

En pratique, lorsque l'on réalise un test statistique :

1. on calcule la p-valeur associée (évite de choisir au préalable le niveau du test),
2. on rejette le test en fixant le niveau du test à posteriori (classiquement $\alpha = 0.05$).

Ici, les procédures de Bonferroni et de BH nécessitent de fixer un seuil α au préalable (correspondant au contrôle du FWER et du FDR). Cependant, une **correction**, dite de Bonferroni ou de BH, permet de corriger les *p*-valeurs initiales pour aboutir à la simple règle de décision :

$$\text{Si } \tilde{p}_{(i)} < \alpha, \text{ alors on rejette } (H_0).$$

Sélection par tests multiples

p-valeurs ajustées (R)

```
pvalue <- c(0.001,0.1,0.52,0.000001,0.34,0.05,0.042)
pvalue_adjust <- p.adjust(pvalue,method = "BH")
round(pvalue_adjust,3)
```

```
## [1] 0.004 0.140 0.520 0.000 0.397 0.088 0.088
```

```
pvalue_adjust <- p.adjust(pvalue,method = "bonferroni")
round(pvalue_adjust,3)
```

```
## [1] 0.007 0.700 1.000 0.000 1.000 0.350 0.294
```

III. 3. Réduction de dimension

Réduction de dimension

Introduction

Une manière d'éviter les écueils de la grande dimension consiste à réduire la dimension du problème en projetant les p variables évoluant dans un espace vectoriel de dimension p dans un sous-espace vectoriel de dimension beaucoup plus petit. Pour cela, on :

1. cherche un sous-espace vectoriel de dimension plus petite que p,
2. remplace le nuage de point initial par sa projection orthogonale sur le sous-espace ainsi défini.

Problématique : choisir le sous-espace de telle sorte que le nouveau nuage soit aussi représentatif que possible des données initiales.

Réduction de dimension

Cadre mathématique

On s'intéresse ici aux méthodes de réduction de dimension par combinaisons linéaires de variables (ACP et PLS).

Objectif : construction de :

- une matrice A de taille $p \times r$ ($r << p$) contenant en colonne les coefficients des combinaisons linéaires des anciennes variables (les vecteurs engendrant le nouvel espace),
- une matrice Z de taille $n \times r$ contenant les r nouvelles variables,

telles que :

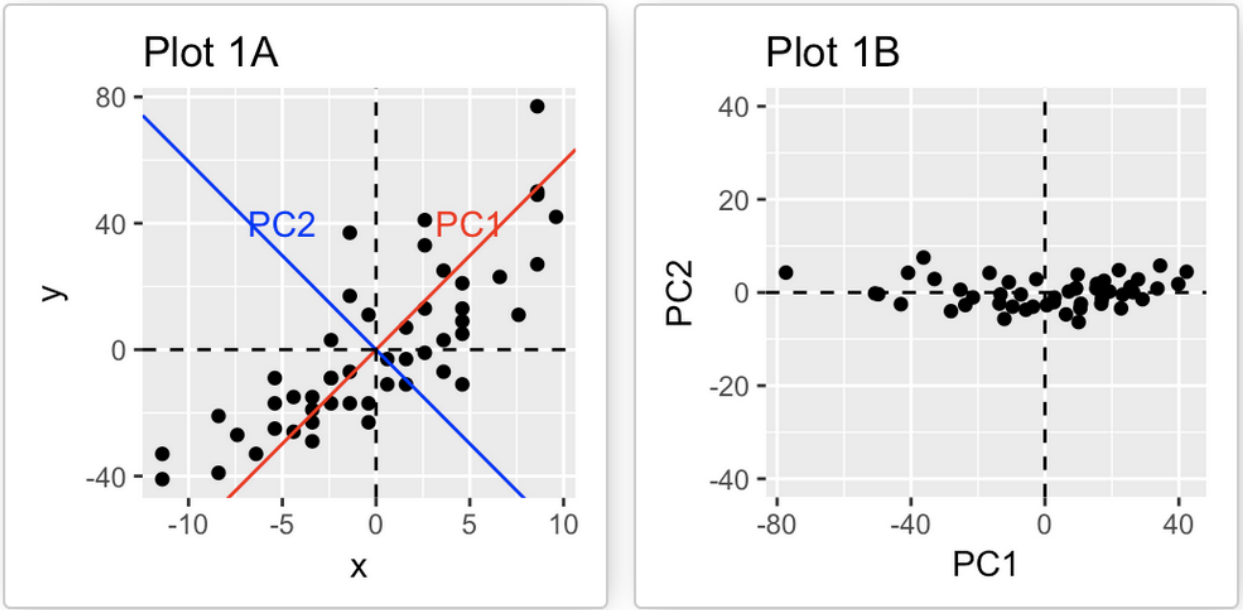
$$Z = XA.$$

Réduction de dimension

Analyse par Composantes Principales (ACP)

Les composantes principales $Z^1, ..., Z^r$ sont construites de telle sorte à garder le plus d'information possible contenue dans $X^1, ..., X^p$: la variance des coordonnées des n individus sur chaque nouvel axe doit être maximale.

Ainsi, le nuage de points se répartit bien sur l'axe, gardant la diversité initiale du nuage, ce qui ne serait pas le cas si tous les points étaient projetés au même endroit (variance nulle).



Réduction de dimension

Analyse par Composantes Principales (ACP)

Etape 0 : normalisation des données

Afin de limiter l'effet de trop "grosses" variables présentes, les variables $X^1, ..., X^p$ sont **normées** :

$$\forall i \in [\![1, p]\!], \quad X^i \sim \mathcal{N}(0, 1).$$

Ceci a pour effet :

- de pouvoir comparer des variables à échelles différentes (*avantage*),
- de lisser le signal (*inconvénient*).

Réduction de dimension

Analyse par Composantes Principales (ACP)

Etape 1 : construction du 1er axe

Le 1er axe Z^1 est choisi comme étant la combinaison linéaire de $X^1, ...X^p$ de variance maximale :

$$Z^1 = X\alpha_1,$$

avec $\|\alpha_1\| = 1$ et $Var(X\alpha_1)$ maximale parmi les vecteurs de la forme $X\alpha$.

- $\alpha_1 \in \mathbb{R}^p$ représente la direction du 1er axe principal,
- $X\alpha_1 \in \mathbb{R}^n$ est l'ensemble des coordonnées du nuage de points sur cet axe.

Etape 2 : construction du 2ième axe

Le 2nd axe Z^2 est choisi comme étant la combinaison linéaire de $X^1, ...X^p$ de variance maximale :

$$Z^2 = X\alpha_2,$$

avec $\|\alpha_2\| = 1$ et $Var(X\alpha_2)$ maximale parmi les vecteurs de la forme $X\alpha$.

On y ajoute la **contrainte** :

$$\langle Z^2, Z^1 \rangle = 0.$$

Réduction de dimension

Analyse par Composantes Principales (ACP)

Etape k : construction du k-ième axe

De manière plus générale, Z^k est choisi comme étant la combinaison linéaire de $X^1, ...X^p$ de variance maximale :

$$Z^k = X\alpha_k,$$

où

$$\alpha_k = \underset{\alpha \in \mathbb{R}^p}{argmax} \quad Var(X\alpha).$$

sous les contraintes :

$$\|\alpha_k\| = 1 \text{ et } \forall \ell \in [\![1, k-1]\!], \quad {}^t\alpha_k {}^tXX\alpha_\ell = 0.$$

Par construction, tous les axes sont orthogonaux et ils sont ordonnés du plus informatif Z^1 au moins informatif Z^r.

Réduction de dimension

Analyse par Composantes Principales (ACP)

Etape k : construction du k-ième axe

De manière plus générale, Z^k est choisi comme étant la combinaison linéaire de $X^1, ...X^p$ de variance maximale :

$$Z^k = X\alpha_k,$$

où

$$\alpha_k = \underset{\alpha \in \mathbb{R}^p}{argmax} \ {}^t\alpha {}^tXX\alpha.$$

sous les contraintes :

$$\|\alpha_k\| = 1 \text{ et } \forall \ell \in [\![1, k-1]\!], \quad {}^t\alpha_k {}^tXX\alpha_\ell = 0.$$

Par construction, tous les axes sont orthogonaux et ils sont ordonnés du plus informatif Z^1 au moins informatif Z^r.

Réduction de dimension

Analyse par Composantes Principales (ACP)

Etape k : construction du k-ième axe

De manière plus générale, Z^k est choisi comme étant la combinaison linéaire de $X^1, ...X^p$ de variance maximale :

$$Z^k = X\alpha_k,$$

où

$$\alpha_k = \underset{\alpha \in \mathbb{R}^p}{argmax} \ {}^t\alpha {}^tXX\alpha = {}^t\alpha\Sigma\alpha,$$

avec Σ la matrice de covariance empirique de X, sous les contraintes :

$$\|\alpha_k\| = 1 \text{ et } \forall \ell \in [\![1, k-1]\!], \quad {}^t\alpha_k {}^tXX\alpha_\ell = 0.$$

Par construction, tous les axes sont orthogonaux et ils sont ordonnés du plus informatif Z^1 au moins informatif Z^r.

Réduction de dimension

Analyse par Composantes Principales (ACP)

En pratique, d'un point de vue algorithmique :

- soit on trouve α_1 puis on projette tous les individus (qui sont des points de $\mathbb{R}^p$) sur $(\alpha_1)^\perp$. On relance alors la résolution du problème d'optimisation pour trouver α_2,...

- soit on utilise le fait que les α_k correspondent aux vecteurs propres de Σ (qui est diagonalisable car symétrique), ordonnés par ordre décroissant de leur valeur propre associée. Ceci permet d'obtenir tous les α_k d'un seul coup!

Réduction de dimension

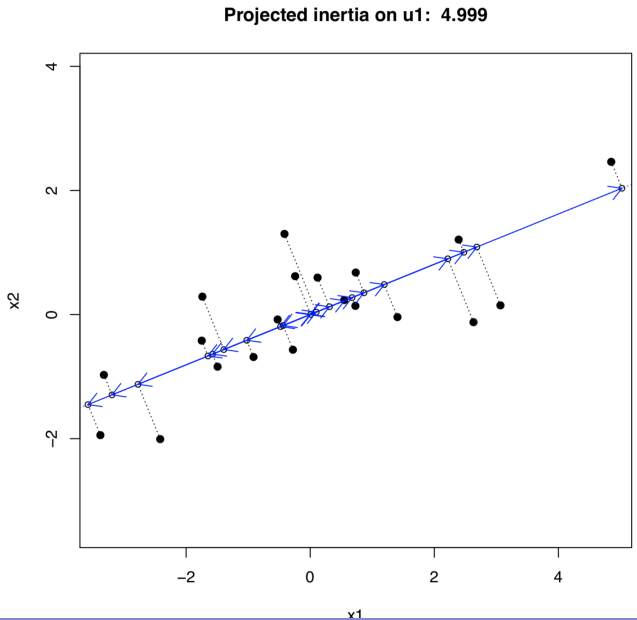
Analyse par Composantes Principales (ACP)

Choix du nombre d'axes : déterminer le nombre r d'axes à retenir est une problématique centrale pour faire de la réduction de dimension. Il existe de nombreux critères basés sur :

- la part d'inertie :

$$r = \underset{k<p}{\text{argmin}} \ \{\mathcal{I}_k > \tau\},$$

où $\mathcal{I}_k$ est l'inertie de la composante k, qui mesure la dispersion des points autour du centre de gravité dans un nuage de points.

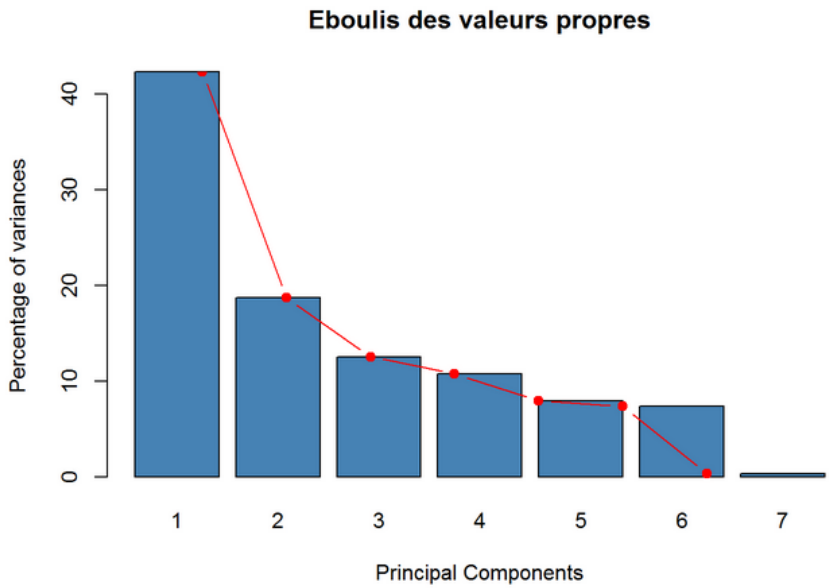


Réduction de dimension

Analyse par Composantes Principales (ACP)

Choix du nombre d'axes : déterminer le nombre r d'axes à retenir est une problématique centrale pour faire de la réduction de dimension. Il existe de nombreux critères basés sur :

- la règle de Kaiser : on ne conserve que les valeurs propres supérieures à leur moyenne.
- l'éboulis des valeurs propres : graphique présentant la décroissance des valeurs propres. On cherche un coude dans le graphe pour déterminer r.

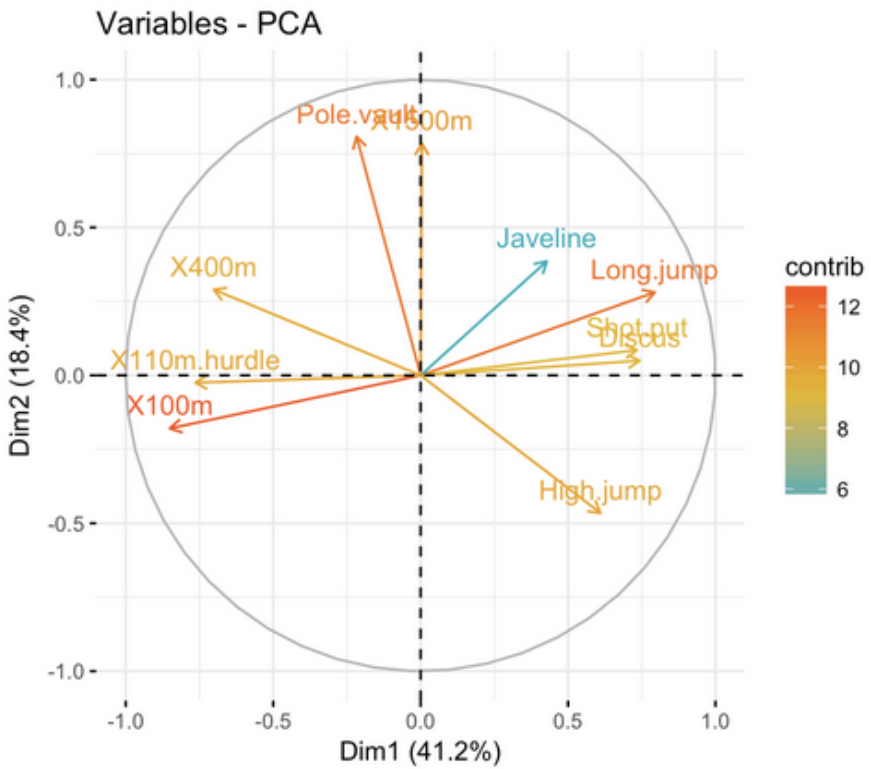


Réduction de dimension

Analyse par Composantes Principales (ACP)

Interprétation de l'ACP :

- Les axes factoriels sont interprétés par rapport aux variables bien représentées en utilisant les contributions ou cercle des corrélations.
- Les contributions des individus permettent d'identifier ceux qui ont une grande influence sur l'ACP. Ces individus sont à étudier parfois séparément.

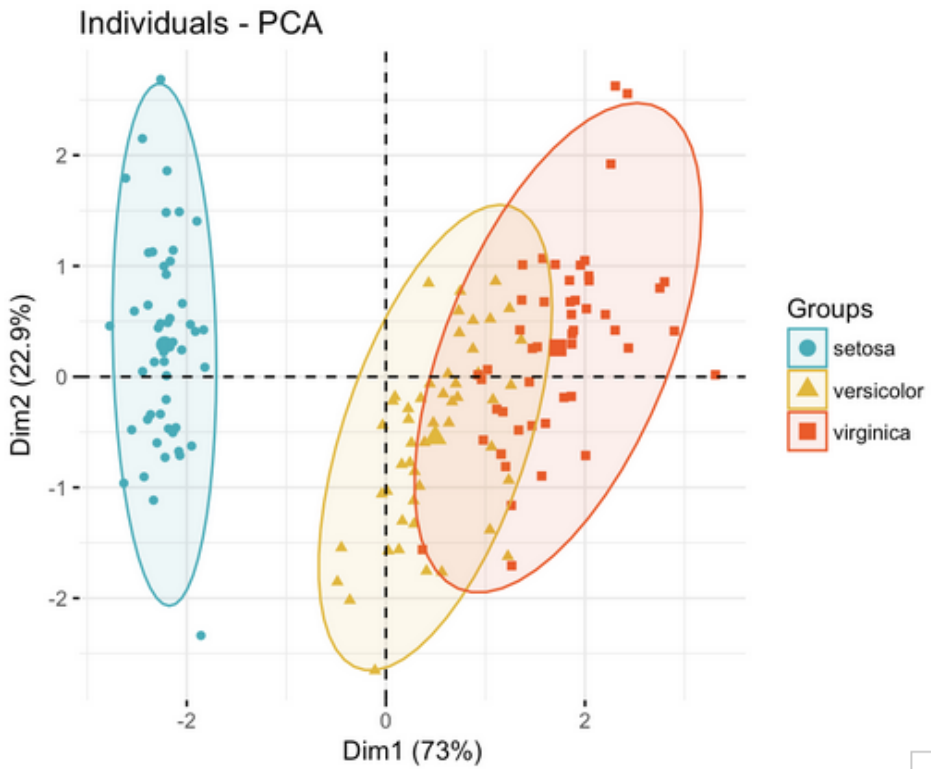


Réduction de dimension

Analyse par Composantes Principales (ACP)

Interprétation de l'ACP :

- Les axes factoriels sont interprétés par rapport aux variables bien représentées en utilisant les contributions ou cercle des corrélations.
- Les contributions des individus permettent d'identifier ceux qui ont une grande influence sur l'ACP. Ces individus sont à étudier parfois séparément.



Réduction de dimension

Régression des moindres carrés partiels (PLS)

Les composantes d'une ACP sont construites pour représenter au mieux l'information contenue dans les variables $X^1, ...X^p$ mais ne tiennent pas compte du lien avec Y, ce qui est le point central d'un modèle de régression.

La PLS est une généralisation de l'ACP au sens où elle construit des variables qui

- représentent au mieux les variables $X^1, ..., X^p$,
- ET sont le plus possible corrélées à Y.

Les composantes de la PLS sont les $Z^1, ..., Z^r$ où, pour tout k, $Z^k = X\alpha_k$ et :

$$\alpha_k = \underset{\alpha \in \mathbb{R}^p}{argmax} \ \ Cov(X\alpha, Y),$$

sous les contraintes :

$$\|\alpha_k\| = 1 \text{ et } \forall \ell \in [\![1, k-1]\!], \quad {}^t\alpha_k {}^tXX\alpha_\ell = 0.$$

Réduction de dimension

Régression des moindres carrés partiels (PLS)

Remarque : on a $Cov(X\alpha, Y)^2 = Corr(X\alpha, Y)^2 Var(X\alpha) Var(Y)$ où *Corr* désigne la corrélation. Les composantes de la PLS sont donc les $Z^1, ..., Z^r$ où, pour tout k, $Z^k = X\alpha_k$ et :

$$\alpha_k = \underset{\alpha \in \mathbb{R}^p}{argmax} \quad Corr(X\alpha, Y)^2 Var(X\alpha),$$

sous les contraintes :

$$\|\alpha_k\| = 1 \text{ et } \forall \ell \in [\![1, k-1]\!], \quad {}^t\alpha_k {}^t X X \alpha_\ell = 0.$$

Ainsi, les composantes choisies :

- maximisent la variance de la projection du nuage de points formé par $X^1, ..., X^p$,
- maximisent la corrélation avec Y.

Réduction de dimension

Régression des moindres carrés partiels (PLS)

En pratique, d'un point de vue algorithmique :

- on peut à nouveau chercher pas à pas en projetant à chaque fois sur l'orthogonal des axes déjà définis,
- ou déterminer la décomposition en valeurs singulières de ${}^tXY{}^tYX$.

Remarque : si Y est multi-dimensionnel, on peut étendre la PLS en cherchant cette fois la meilleure covariance entre une combinaison linéaire des X et une combinaison linéaire des Y :

$$(\alpha_k, \beta_k) = \textit{argmax} \quad \textit{Cov}(X\alpha, Y\beta),$$

sous la contrainte ${}^t\alpha\alpha = 1$ et ${}^t\beta\beta = 1$. On projette ensuite X et Y suivant les vecteurs choisis (déflation) et ainsi de suite.

Réduction de dimension

ACP et PLS dans un cadre de régression

On peut utiliser l'ACP et la PLS comme une étape préliminaire à la régression :

$$Y = \beta_0 + \sum_{j=1}^{p} \beta_j X^j + \varepsilon.$$

Pour cela :

1. on ne considère plus les variables $X^1, ..., X^p$ mais les variables $Z^1, ..., Z^r$,
2. on régresse Y sur ces nouvelles variables $Z^1, ..., Z^r$ ($r < n$, le problème est donc bien défini) :
$$Y = \gamma_0 + \sum_{j=1}^{r} \gamma_j Z^j + \varepsilon,$$
3. on interprète les coefficients à partir des variables initiales suivant :
$$\hat{\beta} = \sum_{j=1}^{r} \hat{\gamma}_j \alpha_j, \quad \text{où } Z^j = X\alpha_j.$$

Réduction de dimension

ACP sur R

Exemple de base :

```
library(FactoMineR)
data(iris)
pca <- PCA(iris[1:4],graph=FALSE) # ACP sur les données
pca$eig # infos sur les vaps
```

```
##        eigenvalue percentage of variance cumulative percentage of
## comp 1 2.91849782              72.9624454
## comp 2 0.91403047              22.8507618
## comp 3 0.14675688               3.6689219
## comp 4 0.02071484               0.5178709
```

```
# (notamment la part de variance expliquée)
```

Réduction de dimension

ACP sur R

Exemple de base :

```
pca$var # ici, infos sur les variables
```

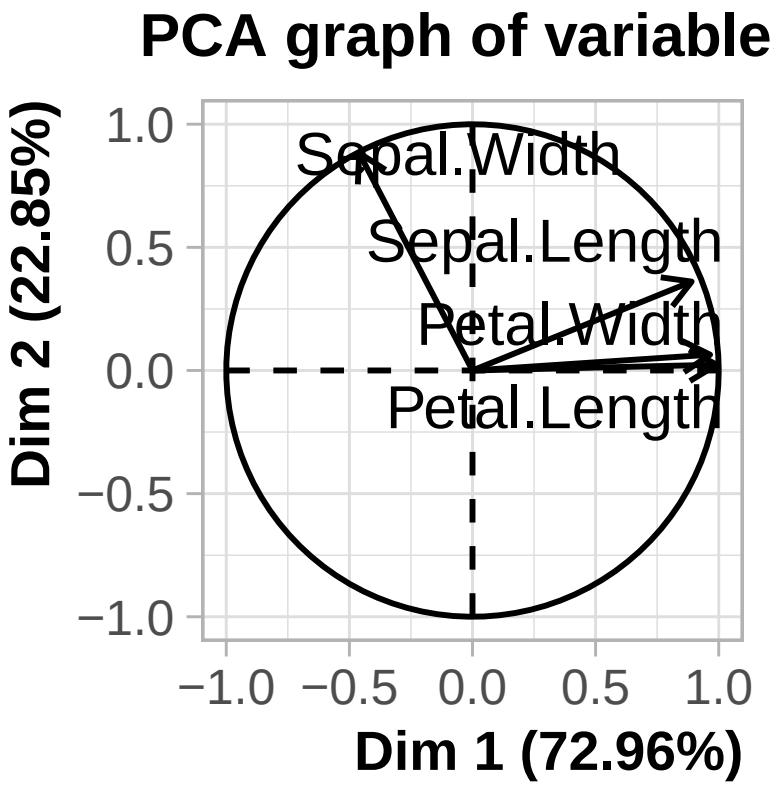
```
## $coord
##                     Dim.1      Dim.2       Dim.3       Dim.4
## Sepal.Length  0.8901688 0.36082989 -0.27565767 -0.03760602
## Sepal.Width  -0.4601427 0.88271627  0.09361987  0.01777631
## Petal.Length  0.9915552 0.02341519  0.05444699  0.11534978
## Petal.Width   0.9649790 0.06399985  0.24298265 -0.07535950
##
## $cor
##                     Dim.1      Dim.2       Dim.3       Dim.4
## Sepal.Length  0.8901688 0.36082989 -0.27565767 -0.03760602
## Sepal.Width  -0.4601427 0.88271627  0.09361987  0.01777631
## Petal.Length  0.9915552 0.02341519  0.05444699  0.11534978
## Petal.Width   0.9649790 0.06399985  0.24298265 -0.07535950
##
## $cos2
```

Réduction de dimension

ACP sur R

Exemple de base :

```
plot(pca, choix="var")
```



Réduction de dimension

ACP et PLS sur R - cadre de régression

Chargement des données :

```
library(pls)
data(gasoline)
str(gasoline)
```

```
## 'data.frame':    60 obs. of  2 variables:
##  $ octane: num  85.3 85.2 88.5 83.4 87.9 ...
##  $ NIR   : 'AsIs' num [1:60, 1:401] -0.0502 -0.0442 -0.0469 -0.04
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr  "1" "2" "3" "4" ...
##   .. ..$ : chr  "900 nm" "902 nm" "904 nm" "906 nm" ...
```

Réduction de dimension

ACP et PLS sur R - cadre de régression

Séparation du jeu de données en deux :

```
# échantillon d'apprentissage
learn <- sample(1:nrow(gasoline),50,replace=FALSE)

# échantillon test
test <- which(!(1:nrow(gasoline)) %in% learn)
gasolinelearn <- gasoline[learn,]
gasolinetest <- gasoline[test,]
dim(gasolinelearn)
```

```
## [1] 50  2
```

```
dim(gasolinetest)
```

```
## [1] 10  2
```

Réduction de dimension

ACP et PLS sur R - cadre de régression

ACP sur l'échantillon d'apprentissage :

```
pcrgasoline <- pcr(octane~NIR,ncomp=10,data=gasolinelearn,
                   scale=TRUE,validation="CV",segments=5)
summary(pcrgasoline)
```

```
## Data:    X dimension: 50 401
##  Y dimension: 50 1
## Fit method: svdpc
## Number of components considered: 10
##
## VALIDATION: RMSEP
## Cross-validated using 5 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps
## CV           1.484    1.443    1.503   0.6929   0.2853   0.2274
## adjCV        1.484    1.435    1.482   0.5464   0.2823   0.2252
##        7 comps  8 comps  9 comps  10 comps
## CV      0.2055   0.2335   0.2265     0.235
## adjCV   0.2049   0.2353   0.2190     0.227
```

Réduction de dimension

ACP et PLS sur R - cadre de régression

PLS sur l'échantillon d'apprentissage :

```
plsgasoline <- plsr(octane~NIR,ncomp=10,data=gasolinelearn,
                    scale=TRUE,validation="CV",segments=5)
summary(plsgasoline)
```

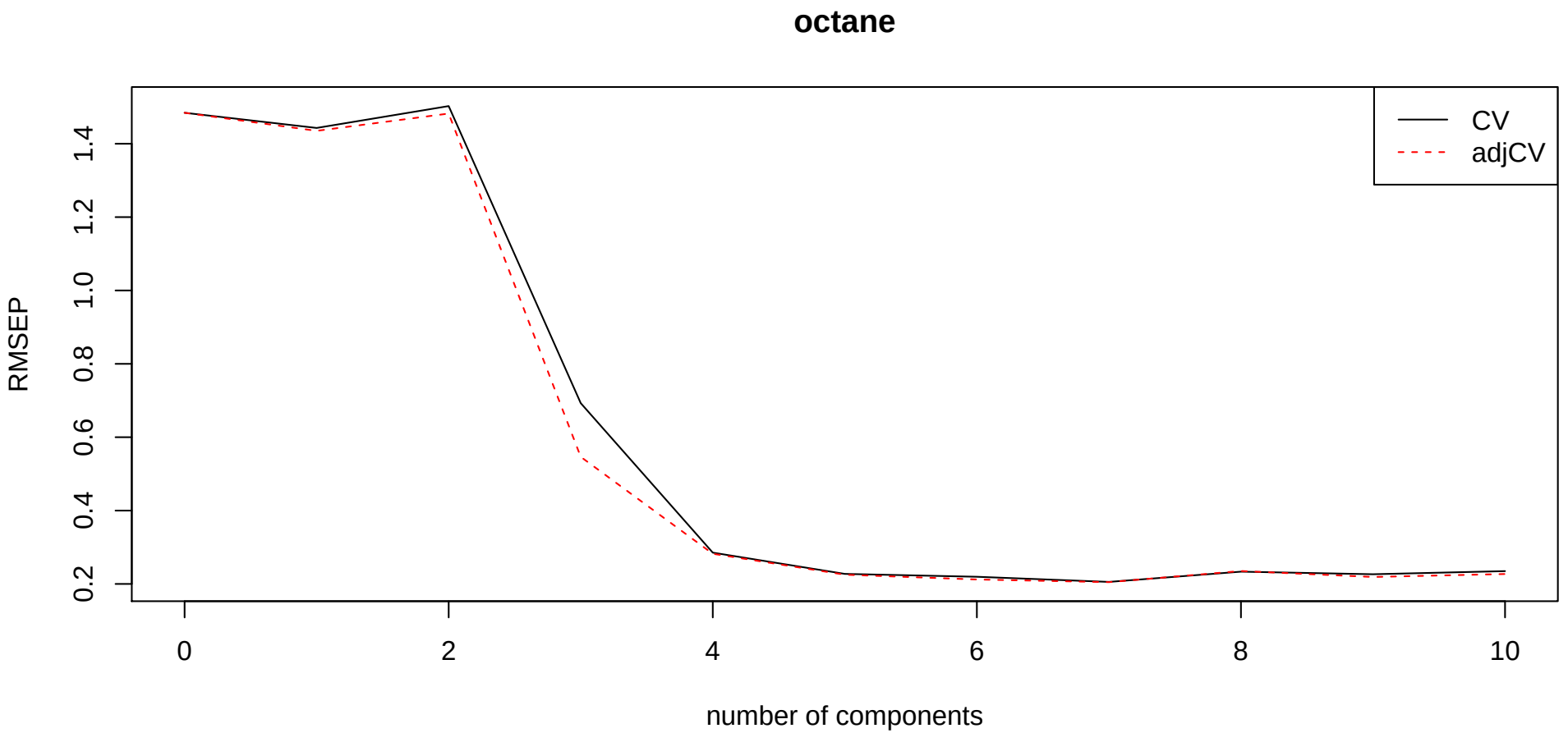
```
## Data:    X dimension: 50 401
##  Y dimension: 50 1
## Fit method: kernelpls
## Number of components considered: 10
##
## VALIDATION: RMSEP
## Cross-validated using 5 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps
## CV           1.484    1.313   0.7391   0.2994   0.2295   0.2106
## adjCV        1.484    1.310   0.7248   0.2857   0.2211   0.2091
##        7 comps  8 comps  9 comps  10 comps
## CV      0.2248   0.2742   0.2738    0.2856
## adjCV   0.2168   0.2550   0.2555    0.2636
```

Réduction de dimension

ACP et PLS sur R - cadre de régression

Erreur de prédiction sur l'échantillon d'apprentissage : permet d'identifier le nombre optimal de composantes.

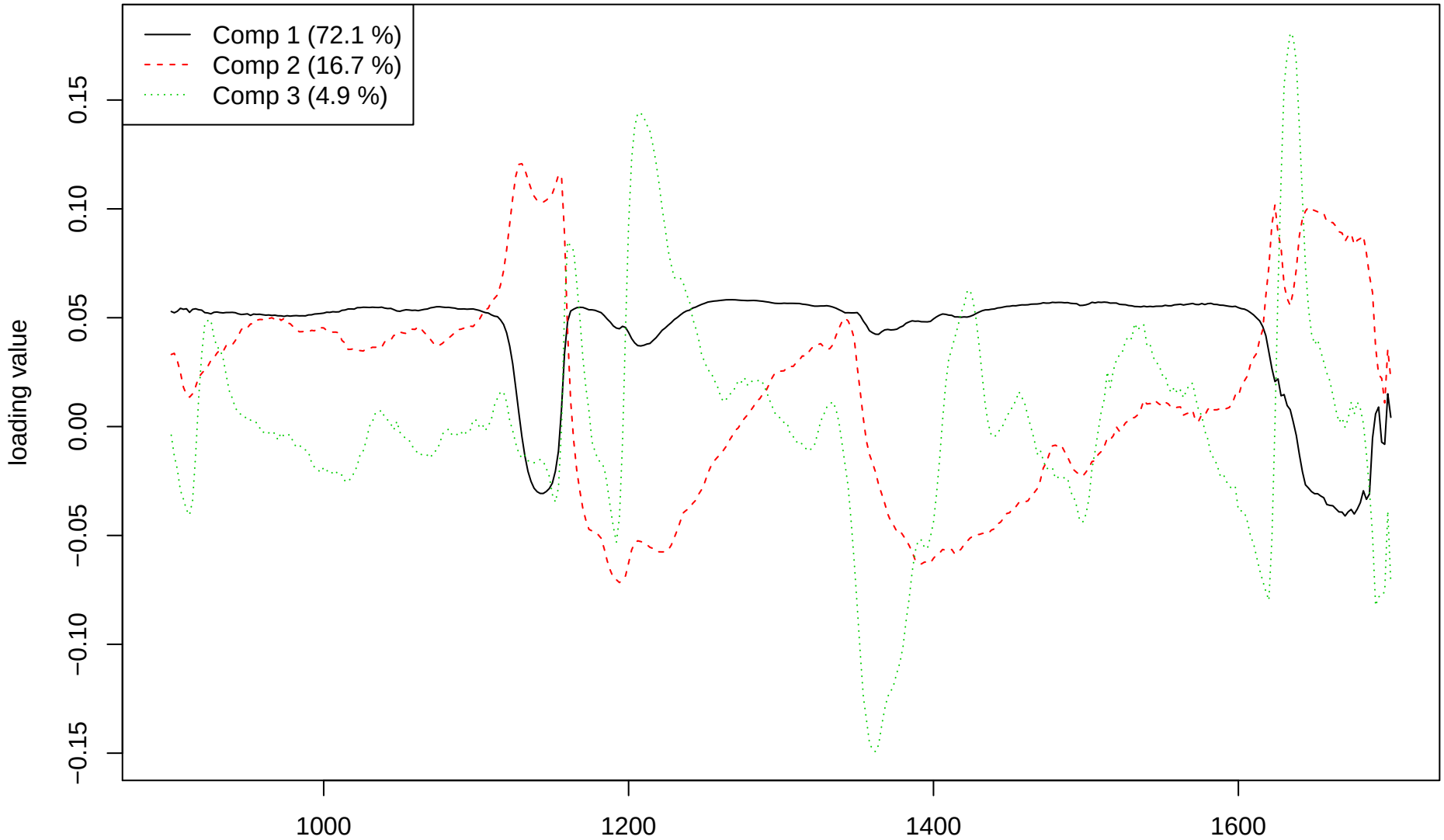
```
plot(RMSEP(pcrgasoline), legendpos = "topright")
```



Réduction de dimension

ACP et PLS sur R - cadre de régression

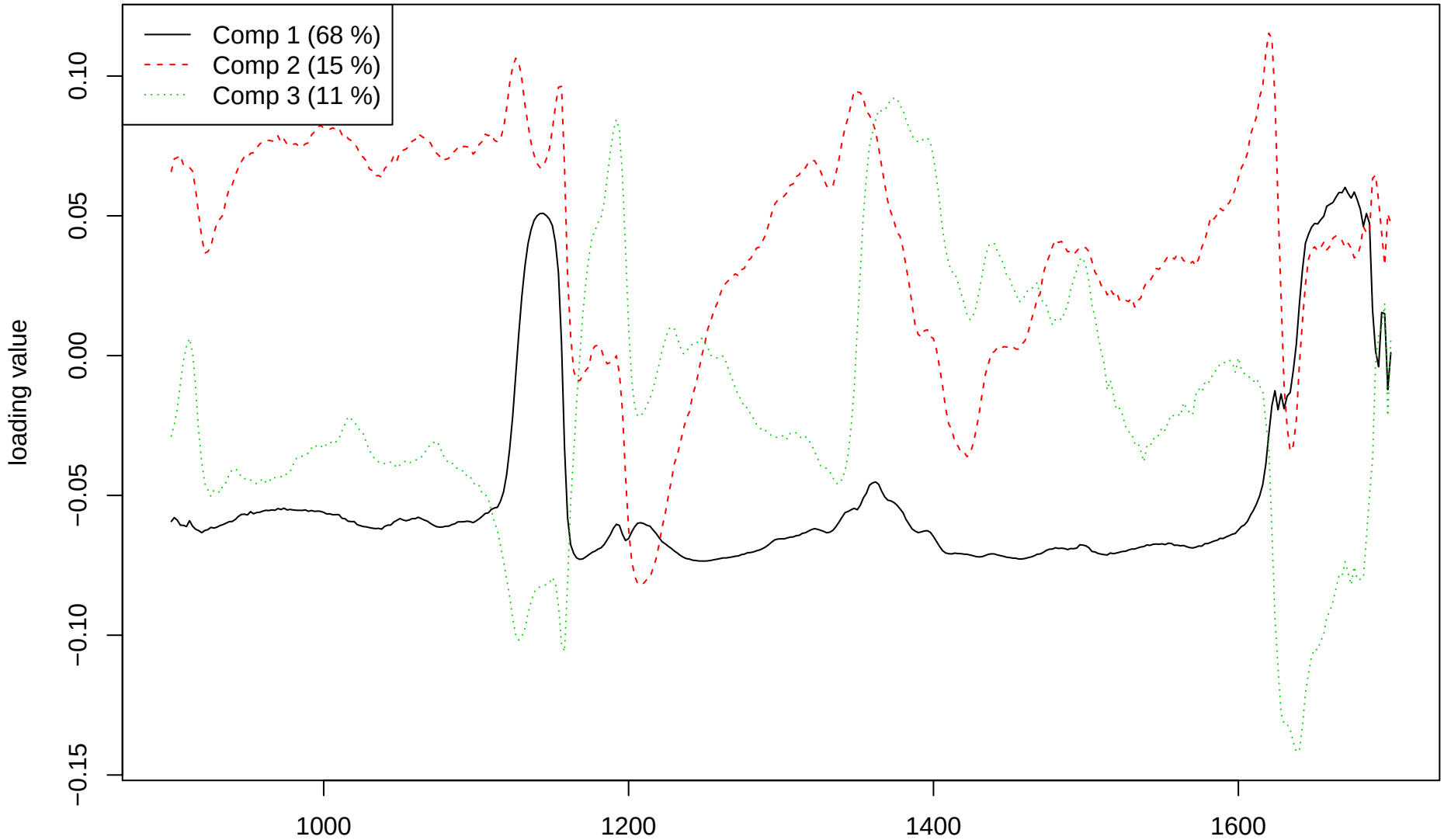
Interprétation des coefficients ACP



Réduction de dimension

ACP et PLS sur R - cadre de régression

Interprétation des coefficients PLS



Réduction de dimension

ACP et PLS sur R - cadre de régression

Erreur d'apprentissage sur l'échantillon test

```
pred <- predict(plsgasoline, ncomp = 3, newdata = gasolinetest)
pred[,,1]
```

```
##        9       12       14       31       32       33       37
## 88.75821 87.74725 88.07812 86.54786 84.55703 84.69913 85.46547 85
##       52       59
## 87.32936 89.13162
```

```
gasolinetest$octane
```

```
##  [1] 88.70 88.25 88.00 86.30 84.40 84.70 85.25 85.30 87.60 89.60
```

```
RMSEP(plsgasoline, newdata = gasolinetest)
```

```
## (Intercept)      1 comps      2 comps      3 comps      4 comps
##      1.8069       1.5542       0.5247       0.2685       0.1888
##      5 comps      6 comps      7 comps      8 comps      9 comps
```

III. 4. Sélection de variables par pénalisation

Sélection de variables par pénalisation

Introduction

On se place dans le cadre de la régression linéaire (gaussienne ou logistique) en grande dimension ($p > n$). La matrice ${}^tXX$ n'est alors pas forcément de rang p et n'est plus inversible, rendant l'estimateur des moindres carrés incalculable. De même, en présence de variables fortement corrélées, la matrice peut être inversible mais son inverse, et donc l'estimateur des moindres carrés, est très instable.

Deux problématiques se posent alors :

- comment limiter les effets des corrélations?
- comment choisir un nombre restreint de variables qui agissent sur la réponse?

Sélection de variables par pénalisation

Principe

On considère le modèle de régression linéaire :

$$Y = X\beta + \varepsilon.$$

On note $\ell(\beta, X, Y)$ la vraisemblance associée à une valeur β au vu des données X et Y. Afin de favoriser un certain comportement de la solution, on peut remplacer l'estimateur du maximum de vraisemblance :

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^p}{\textit{argmax}} \ \ \ell(\beta, X, Y)$$

par

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^p}{\textit{argmax}} \ \ \ell(\beta, X, Y) - \lambda \textit{pen}(\beta),$$

où $\textit{pen}(\beta)$ est une fonction de pénalité à choisir. La valeur de λ (*paramètre de pénalisation*) fixe le degré de pénalité que l'on veut considérer.

Sélection de variables par pénalisation

Régression Ridge

L'estimateur **Ridge** est défini par :

$$\hat{\beta}_{ridge} = \underset{\beta \in \mathbb{R}^p}{argmax} \ \ell(\beta, X, Y) - \lambda \|\beta\|_2^2,$$

ou, de façon équivalente dans le cas gaussien,

$$\hat{\beta}_{ridge} = \underset{\beta \in \mathbb{R}^p}{argmin} \ \|Y - X\beta\|_2^2 + \lambda \|\beta\|_2^2.$$

Il vaut alors :

$$\hat{\beta}_{ridge} = ({}^tXX + \lambda I)^{-1\,t}XY.$$

- La pénalité ridge est utilisée pour diminuer la grande variance induite sur β par la présence de variables corrélées.
- L'estimateur ridge est biaisé contrairement à celui des moindres carrés.

Sélection de variables par pénalisation

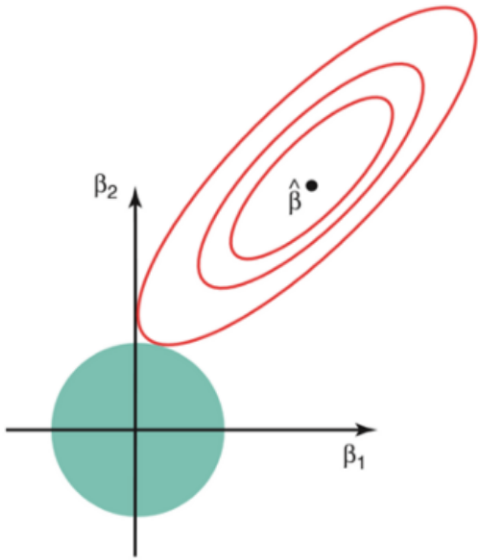
Régression Ridge - problème dual

Le problème de minimisation

$$\hat{\beta}_{ridge} = \underset{\beta \in \mathbb{R}^p}{argmin} \ \|Y - X\beta\|_2^2 + \lambda \|\beta\|_2^2,$$

est équivalent à un problème dual de la forme :

$$\hat{\beta}_{ridge} = \underset{\beta \in \mathbb{R}^p}{argmin} \ \|Y - X\beta\|_2^2 \ \text{ sous la contrainte } \sum_i \beta_i^2 \leq c(\lambda).$$



Sélection de variables par pénalisation

Régression Lasso

L'estimateur **Lasso** est défini par :

$$\hat{\beta}_{lasso} = \underset{\beta \in \mathbb{R}^p}{argmax} \ \ell(\beta, X, Y) - \lambda \|\beta\|_1,$$

ou, de façon équivalente dans le cas gaussien,

$$\hat{\beta}_{lasso} = \underset{\beta \in \mathbb{R}^p}{argmin} \ \|Y - X\beta\|_2^2 + \lambda \|\beta\|_1.$$

- La pénalité lasso est utilisée pour obtenir des solutions parcimonieuses, c'est-à-dire telles que beaucoup de coefficients soient nuls (plus λ est grand, plus les solutions sont parcimonieuses).
- L'estimateur lasso est en général un estimateur de grande variance avec des problèmes de stabilité, notamment en présence de variables corrélées.
- L'estimateur Lasso n'a pas d'écriture propre, il faut le déterminer par un algorithme d'optimisation (*Lars*).

Sélection de variables par pénalisation

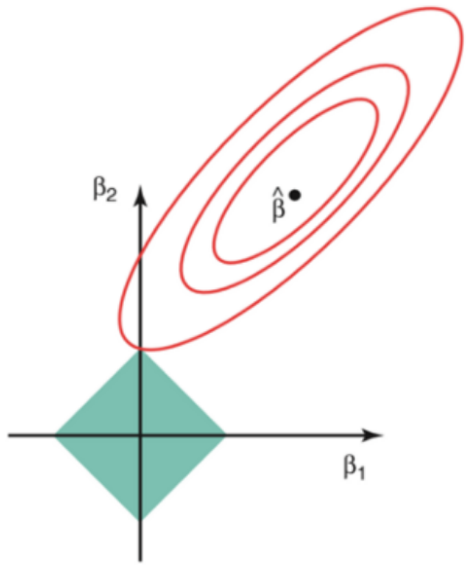
Régression Lasso - problème dual

Le problème de minimisation

$$\hat{\beta}_{lasso} = \underset{\beta \in \mathbb{R}^p}{argmin} \ \|Y - X\beta\|_2^2 + \lambda \|\beta\|_1,$$

est équivalent à un problème dual de la forme :

$$\hat{\beta}_{lasso} = \underset{\beta \in \mathbb{R}^p}{argmin} \ \|Y - X\beta\|_2^2 \ \text{ sous la contrainte } \sum_i |\beta_i| \leq c(\lambda).$$



Sélection de variables par pénalisation

Variante Elastic-Net

L'estimateur **Elastic-Net** est défini par :

$$\hat{\beta}_{EN} = \underset{\beta \in \mathbb{R}^p}{\textit{argmax}} \;\; \ell(\beta, X, Y) - \lambda \left(\alpha \|\beta\|_1 + (1-\alpha)\|\beta\|_2^2 \right).$$

- Avantage : l'estimateur Elastic-Net est un juste milieu entre pénalité Lasso et Ridge.
- Inconvénient : l'estimateur Elastic-Net nécessite de calibrer deux paramètres.

Sélection de variables par pénalisation

Variante Group-Lasso

On considère que les variables sont réparties dans K groupes prédéfinis. On note β_k les coordonnées du vecteur β correspondant aux variables du groupe k.
L'estimateur **Group-Lasso** est défini par :

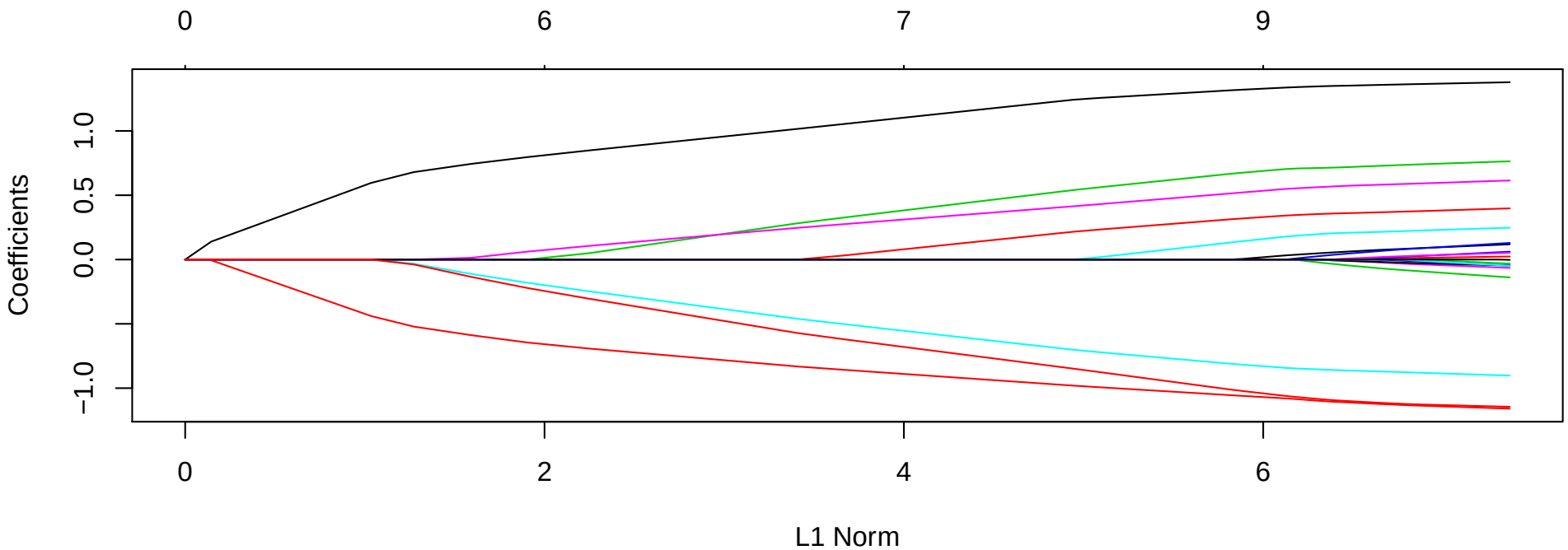
$$\hat{\beta}_{GL} = \underset{\beta \in \mathbb{R}^p}{argmax} \ \ \ell(\beta, X, Y) - \sum_{k=1}^{K} \lambda_k \|\beta_k\|_2.$$

- Avantage : l'estimateur Group-Lasso est un juste milieu entre pénalité Lasso et Ridge : les corrélations sont prises à l'intérieur des groupes et de nombreux groupes sont annulés entièrement.
- Inconvénient : les groupes doivent être déterminés à l'avance.

Sélection de variables par pénalisation

Applications sur R

```
library(glmnet)
data(QuickStartExample) # jeu de données du package

fit = glmnet(x, y) # met en place le modèle Lasso
plot(fit)
```



Sélection de variables par pénalisation

Applications sur R

```
print(fit)
```

```
##
## Call:  glmnet(x = x, y = y)
##
##       Df    %Dev   Lambda
##  [1,]  0 0.00000 1.631000
##  [2,]  2 0.05528 1.486000
##  [3,]  2 0.14590 1.354000
##  [4,]  2 0.22110 1.234000
##  [5,]  2 0.28360 1.124000
##  [6,]  2 0.33540 1.024000
##  [7,]  4 0.39040 0.933200
##  [8,]  5 0.45600 0.850300
##  [9,]  5 0.51540 0.774700
## [10,]  6 0.57350 0.705900
## [11,]  6 0.62550 0.643200
## [12,]  6 0.66870 0.586100
```

Sélection de variables par pénalisation

Applications sur R

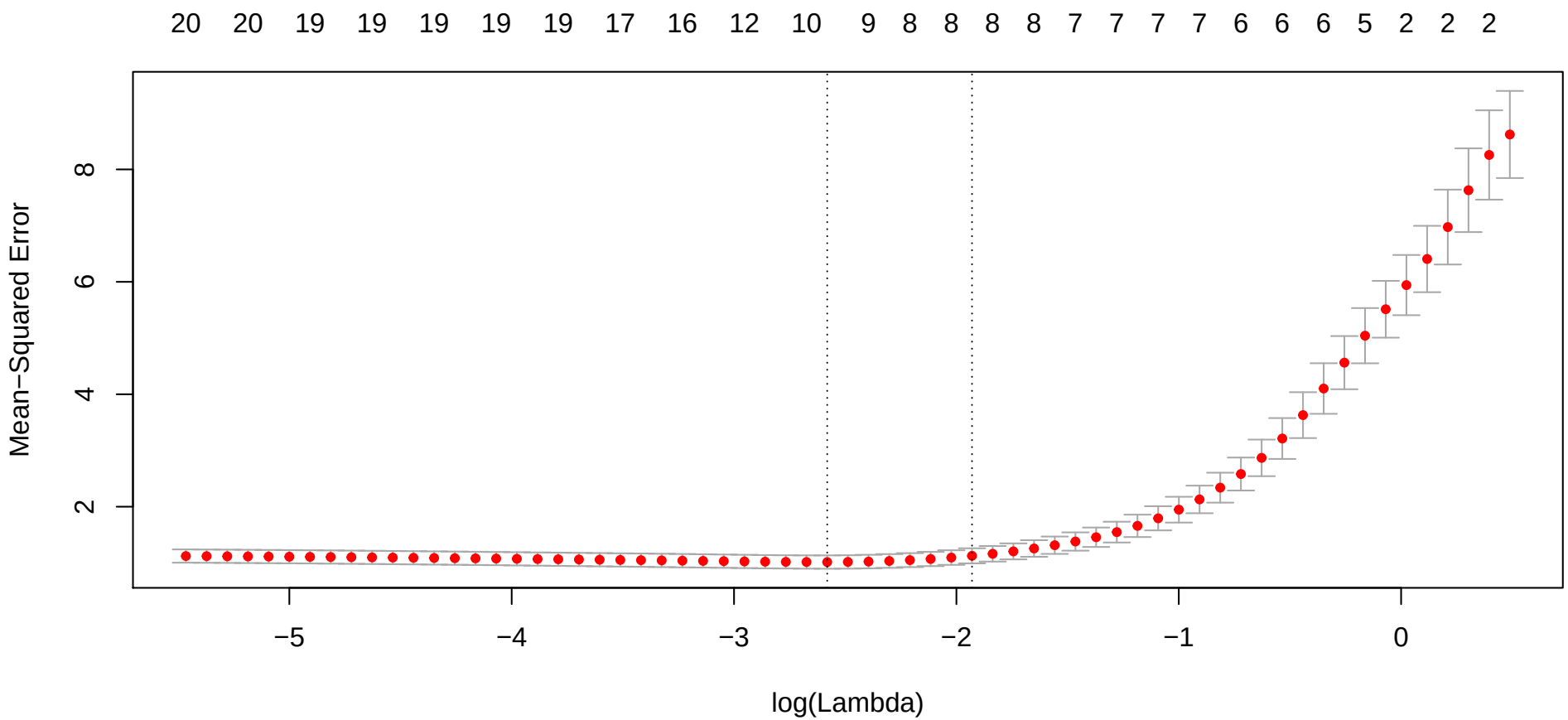
```
coef(fit)
```

```
## 21 x 67 sparse Matrix of class "dgCMatrix"

##    [[ suppressing 67 column names 's0', 's1', 's2' ... ]]

##
## (Intercept) 0.6607581  0.631235043  0.5874616  0.5475769  0.51123
## V1          .          0.139264992  0.2698292  0.3887945  0.49719
## V2          .          .            .          .          .
## V3          .          .            .          .          .
## V4          .          .            .          .          .
## V5          .          .            .          .          .
## V6          .          .            .          .          .
## V7          .          .            .          .          .
## V8          .          .            .          .          .
## V9          .          .            .          .          .
## V10         .          .            .          .          .
## V11
```

Sélection de variables par pénalisation

Applications sur R

```
cvfit = cv.glmnet(x, y)
plot(cvfit)
```



Sélection de variables par pénalisation

Applications sur R

```
coef(cvfit, s = "lambda.min")
```

```
## 21 x 1 sparse Matrix of class "dgCMatrix"
##                       1
## (Intercept)  0.14867414
## V1           1.33377821
## V2           .
## V3           0.69787701
## V4           .
## V5          -0.83726751
## V6           0.54334327
## V7           0.02668633
## V8           0.33741131
## V9           .
## V10          .
## V11          0.17105029
## V12          .
## V13          .
```

Sélection de variables par pénalisation

Applications sur R

Les autres modèles peuvent être définis à l'aide de la fonction `glmnet()` en spécifiant la valeur du paramètre `alpha` :

```
# met en place le modèle Ridge
fit_ridge = cv.glmnet(x, y, alpha= 0)
# met en place le modèle Elastic Net
fit_EN = cv.glmnet(x, y, alpha= 0.8)
length(which(abs(coef(fit_ridge, s = "lambda.min"))>0))
```

```
## [1] 21
```

```
length(which(abs(coef(fit_EN, s = "lambda.min"))>0))
```

```
## [1] 10
```