Projet STA203 :

Rapport

Raphaël BERNAS
Mayeul DOCQ

mai 2024

Table des matières

Introduction

L'entièreté de ce projet est réalisée en R.

Les jeux de donnée utilisé pour ce projet sont les jeux gasolineTrain et gasolineTest issus du package $Rpls$

L'objectif de ce projet est de développer et mettre en œuvre des méthodes efficaces pour traiter des données de grande dimension en utilisant des techniques de pénalisation robustes pour réduire le biais (induit par la pénalisation). Il s'agira donc d'explorer et de mettre en pratique des approches permettant de modéliser ces données volumineuses tout en atténuant les effets de biais.

1 De la théorie

1.1 Question 1 : Une régression linéaire ?

Lorsqu'on modélise une régression linéaire de la forme :

$$Y = \theta_0 \mathbf{1}_n + X\theta + \varepsilon$$

où :
— $Y \in \mathbb{R}^n$ est le vecteur des observations,
— $\theta_0 \in \mathbb{R}$ est le biais,
— $\mathbf{1}_n$ est un vecteur colonne de dimensions $n \times 1$ contenant uniquement des uns,
— X est la matrice des prédicteurs de dimension $n \times p$,
— $\theta \in \mathbb{R}^p$ est le vecteur des coefficients,
— $\varepsilon \in \mathbb{R}^n$ est le vecteur des erreurs,
nous devons considérer plusieurs risques associés aux données de grande dimension :

1. **Surajustement :** Avec un grand nombre de prédicteurs par rapport au nombre d'observations, il existe un risque élevé de surajustement, où le modèle s'adapte trop étroitement aux données d'apprentissage et ne généralise pas bien aux nouvelles données. Pour atténuer ce risque, des techniques de régularisation comme la régression ridge ou la régression LASSO peuvent être utilisées pour pénaliser les coefficients et réduire leur magnitude.
2. **Forte variance des coefficients (θ) :** Dans les données de grande dimension, les estimations des coefficients peuvent avoir une forte variance, ce qui signifie qu'ils peuvent varier considérablement d'un échantillon d'apprentissage à un autre. Cela rend les interprétations des coefficients moins fiables et peut rendre le modèle instable. La régularisation peut également aider à réduire la variance des coefficients.
3. **Multicolinéarité :** Avec un grand nombre de prédicteurs, il peut y avoir des problèmes de multicolinéarité, où certains prédicteurs sont fortement corrélés entre eux. Cela peut rendre les estimations des coefficients instables et difficiles à interpréter.

Dans le cadre de la régression ridge, nous cherchons à minimiser la fonction de perte suivante :

$$\text{argmin}_\theta \left\{ \| Y - \theta_0 \mathbf{1} - X\theta \|_2^2 + \lambda \| \theta \|_2^2 \right\}$$

où :

— $Y \in \mathbb{R}^n$ est le vecteur des observations,
— $\theta_0 \in \mathbb{R}$ est le biais,
— $\mathbf{1}$ est un vecteur colonne de dimensions $n \times 1$ contenant uniquement des uns,
— X est la matrice des prédicteurs de dimension $n \times p$,
— $\theta \in \mathbb{R}^p$ est le vecteur des coefficients,
— λ est le paramètre de régularisation (aussi appelé hyperparamètre), contrôlant la force de la régularisation.

L'atout de la régression ridge dans cette situation de données de grande dimension est sa capacité à réduire la variance des coefficients (θ) en imposant une pénalité sur leur norme ℓ_2. Cela permet de stabiliser les estimations des coefficients, ce qui est particulièrement bénéfique lorsque le nombre de prédicteurs est très élevé par rapport au nombre d'observations. En contrôlant le paramètre de régularisation λ, on peut trouver un compromis approprié entre ajustement aux données d'apprentissage et généralisation aux nouvelles données.

1.2 Question 2 : Estimateur de Ridge

Lorsque l'intercept n'est pas pénalisé dans la régression ridge, cela signifie que seul le terme de pénalité est ajouté aux coefficients des variables explicatives, pas à celui de l'intercept.

La forme de l'estimateur des coefficients dans ce cas peut être démontrée comme suit :

Nous avons la fonction de perte pour la régression ridge sans pénalité de l'intercept :

$$L(\theta) = \| Y - \theta_0 \mathbf{1} - X\theta \|_2^2 + \lambda \left(\|\theta\|_2^2 - \delta \right)$$

Pour trouver l'estimateur des coefficients, nous minimisons cette fonction de perte par rapport à θ. En dérivant par rapport à θ, nous obtenons :

$$\frac{\partial L(\theta)}{\partial \theta} = -2X^T(Y - \theta_0 \mathbf{1} - X\theta) + 2\lambda\theta$$

En réglant cela à zéro pour trouver le minimum, nous avons :

$$X^T(Y - \theta_0 \mathbf{1} - X\theta) = \lambda\theta$$

$$X^T Y - \theta_0 X^T \mathbf{1} - X^T X\theta = \lambda\theta$$
$$X^T Y - \theta_0 X^T \mathbf{1} = (X^T X + \lambda I_p)\theta$$

En résolvant pour θ, nous obtenons :

$$\hat{\theta} = (X^T X + \lambda I_p)^{-1}(X^T Y - \theta_0 X^T \mathbf{1})$$

$$\boxed{\hat{\theta} = (X^T X + \lambda I_p)^{-1} X^T (Y - \theta_0 \mathbf{1})}$$

où :
— $Y \in \mathbb{R}^n$ est le vecteur des observations,
— X est la matrice des prédicteurs de dimension $n \times p$,
— λ est le paramètre de régularisation,
— I_p est la matrice identité de dimension $p \times p$.

Ce qui correspond à la forme de l'estimateur des coefficients sans l'intercept. On dérive maintenant par rapport a θ_0 :

$$\frac{\partial L(\theta_0)}{\partial \theta_0} = -2\mathbb{1}^T(Y - \theta_0 \mathbb{1} - X\theta)$$

D'où :

$$\mathbb{1}^T(Y - \theta_0\mathbb{1} - X\theta) = 0$$

$$\boxed{\hat{\theta_0} = \frac{1}{n}\mathbb{1}^T(Y - X\theta)}$$

On peut alors simplifier θ_0 dans la formule de $\hat{\theta}$:

$$\hat{\theta} = (X^T X + \lambda I_p)^{-1} X^T (Y - \frac{1}{n}\mathbb{1}\mathbb{1}^T(Y - X\hat{\theta}))$$

La relation entre la paramétrisation $\tilde{\theta}$ lorsque les variables explicatives ont été préalablement centrées et θ lorsque elles ne l'ont pas été peut être exprimée comme suit :

Lorsque les variables explicatives ont été centrées, nous définissons $\tilde{X}$ comme la matrice des prédicteurs centrés, c'est-à-dire chaque colonne de X moins sa moyenne. Mathématiquement, cela peut être écrit comme :

$$\tilde{X} = X - \bar{X}\mathbf{1}_n^T$$

Où $\bar{X}$ est le vecteur des moyennes des variables explicatives et $\mathbf{1}_n$ est un vecteur colonne de dimensions $n \times 1$ contenant uniquement des uns.

L'estimateur de Ridge pour $\tilde{\theta_0}$ est alors donné par :

$$\boxed{\tilde{\theta_0} = \bar{Y}}$$

Donc l'estimateur de Ridge pour $\tilde{\theta}$ est alors :

$$\hat{\tilde{\theta}} = (\tilde{X}^T \tilde{X} + \lambda I_p)^{-1} \tilde{X}^T (Y - \bar{Y}\mathbf{1}_n^T)$$

On pose $\tilde{Y} = Y - \bar{Y}\mathbf{1}_n^T$
En développant cette expression, nous obtenons :

$$\hat{\tilde{\theta}} = (\tilde{X}^T \tilde{X} + \lambda I_p)^{-1} \tilde{X}^T \tilde{Y}$$

$$= \left((X - \bar{X}\mathbf{1}_n^T)^T (X - \bar{X}\mathbf{1}_n^T) + \lambda I_p\right)^{-1} (X - \bar{X}\mathbf{1}_n^T)^T \tilde{Y}$$

$$= \left(X^T X - X^T \bar{X}\mathbf{1}_n^T - \bar{X}\mathbf{1}_n X^T + \bar{X}\mathbf{1}_n^T \bar{X}\mathbf{1}_n^T + \lambda I_p\right)^{-1} (X - \bar{X}\mathbf{1}_n^T)^T \tilde{Y}$$

En utilisant les propriétés de l'algèbre linéaire, nous pouvons simplifier cette expression. Le terme $X^T \bar{X}\mathbf{1}_n^T$ peut être réécrit comme $\bar{X}\mathbf{1}_n^T \bar{X}$. En outre, le terme $\bar{X}\mathbf{1}_n X^T$ est égal à $(\bar{X}\mathbf{1}_n^T X)^T$, qui est équivalent à $X^T \bar{X}\mathbf{1}_n^T$. En utilisant ces simplifications, l'expression devient :

$$\hat{\tilde{\theta}} = \left(X^T X - 2X^T \bar{X}\mathbf{1}_n^T + \bar{X}\mathbf{1}_n^T \bar{X} + \lambda I_p\right)^{-1} (X - \bar{X}\mathbf{1}_n^T)^T \tilde{Y}$$

Maintenant, rappelons que la matrice $X^T X$ est la même dans les deux formulations (centrée et non centrée), car elle dépend uniquement des variables explicatives. La différence réside dans le terme $X^T \bar{X}\mathbf{1}_n^T$. Lorsque les variables explicatives ne sont pas centrées, ce terme est nul. Par conséquent, nous avons :

$$\hat{\tilde{\theta}} = (X^T X + \lambda I_p)^{-1} (X - \bar{X}\mathbf{1}_n^T)^T \tilde{Y}$$

$$\boxed{\hat{\tilde{\theta}} = (X^T X + \lambda I_p)^{-1} X^T \tilde{Y}}$$

Finalement, pour rétablir la relation avec θ, nous avons :

$$\hat{\tilde{\theta}} = (X^T X + \lambda I_p)^{-1}(X^T Y - X^T \bar{Y} \mathbf{1}_n^T)$$

Où $\bar{Y}$ est la moyenne des observations. Donc :

$$\boxed{\hat{\tilde{\theta}} = \hat{\theta} - (X^T X + \lambda I_p)^{-1} n \bar{Y} \bar{X}^T}$$

Attention, ici, il n'y a pas de problème de dimension car $\bar{X}$ est un vecteur des moyennes des colonnes de X là où $\bar{Y}$ est un réel.

1.3 Question 3 : Un problème singulier

Soit X une matrice de dimension $n \times p$. Nous avons la décomposition en valeurs singulières de X :

$$X = U \Sigma V^T$$

où U est une matrice orthogonale $n \times r$, Σ est une matrice diagonale $r \times r$ avec les valeurs singulières $\sigma_1, \sigma_2, \ldots, \sigma_r$ sur la diagonale, et V est une matrice orthogonale $p \times r$. Nous pouvons réécrire cela comme :

$$\begin{aligned} X^T X &= (U \Sigma V^T)^T (U \Sigma V^T) \\ &= V \Sigma^T U^T U \Sigma V^T \\ &= V \Sigma^T \Sigma V^T \end{aligned}$$

Puisque U est orthogonale, $U^T U$ est une matrice identité $r \times r$. Donc :

$$\begin{aligned} X^T X &= V \Sigma^T \Sigma V^T \\ &= V \begin{pmatrix} \sigma_1^2 & 0 & \ldots & 0 \\ 0 & \sigma_2^2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \sigma_r^2 \end{pmatrix} V^T \\ &= \sum_{j=1}^{r} \sigma_j^2 v_j v_j^T \end{aligned}$$

Où v_j est la colonne j de V, et $v_j v_j^T$ est le produit extérieur de v_j par lui-même, formant ainsi une matrice de rang 1.
Nous exprimons $(X^T X + \lambda I_p)^{-1}$ en fonction de la décomposition en valeurs singulières de X comme suit :

$$(X^T X + \lambda I_p)^{-1} = \left(\sum_{j=1}^{r} \sigma_j^2 v_j v_j^T + \lambda I_p \right)^{-1}$$

En utilisant la formule de Sherman-Morrison (cas particulier de Woodbury) pour inverser une somme de matrices, nous avons :

$$= \frac{1}{\lambda} I_p - \sum_{j=1}^{r} \frac{\frac{\sigma_j^2}{\lambda^2}}{\frac{\sigma_j^2}{\lambda} + 1} v_j v_j^T$$

$$= \frac{1}{\lambda}\left(I_p - \sum_{j=1}^{r} \frac{\sigma_j^2}{\sigma_j^2+\lambda} v_j v_j^T\right)$$

$$= \frac{1}{\lambda}\left(\sum_{j=1}^{p} \frac{\sigma_j^2+\lambda}{\sigma_j^2+\lambda} v_j v_j^T - \sum_{j=1}^{r} \frac{\sigma_j^2}{\sigma_j^2+\lambda} v_j v_j^T\right)$$

$$= \frac{1}{\lambda}\left(\sum_{j=r+1}^{p} \frac{\sigma_j^2+\lambda}{\sigma_j^2+\lambda} v_j v_j^T + \sum_{j=1}^{r} \frac{\lambda}{\sigma_j^2+\lambda} v_j v_j^T\right)$$

$$= \sum_{j=r+1}^{p} \frac{1}{\lambda} v_j v_j^T + \sum_{j=1}^{r} \frac{1}{\sigma_j^2+\lambda} v_j v_j^T$$

On en déduit alors que :

$$\begin{aligned}
A_0 &= \lim_{\lambda\to 0}(X^T X + \lambda I_p)^{-1} X^T \\
&= \lim_{\lambda\to 0}\left(\sum_{j=1}^{r} \frac{1}{\sigma_j^2+\lambda} v_j v_j^T + \sum_{j=r+1}^{p} \frac{1}{\lambda} v_j v_j^T\right) X^T \\
&= \sum_{j=1}^{r} \lim_{\lambda\to 0}\left(\frac{1}{\sigma_j^2+\lambda} v_j v_j^T\right) X^T + \sum_{j=r+1}^{p} \lim_{\lambda\to 0}\left(\frac{1}{\lambda} v_j v_j^T\right) X^T \\
&= \sum_{j=1}^{r} \frac{1}{\sigma_j^2} v_j v_j^T X^T + \sum_{j=r+1}^{p} \lim_{\lambda\to 0}\left(\frac{1}{\lambda} v_j v_j^T\right) X^T \\
&= \sum_{j=1}^{r} \frac{1}{\sigma_j^2} v_j v_j^T X^T + \sum_{j=r+1}^{p} 0 \\
&= \sum_{j=1}^{r} \frac{1}{\sigma_j^2} v_j v_j^T X^T
\end{aligned}$$

or $X = U\Sigma V^T$ donc :

$$A_0 = \sum_{j=1}^{r}\sum_{j=1}^{r} \frac{\sigma_j}{\sigma_j^2} v_j v_j^T v_j u_j^T$$

or $v_j^T v_i = \delta_{i,j}$ donc :

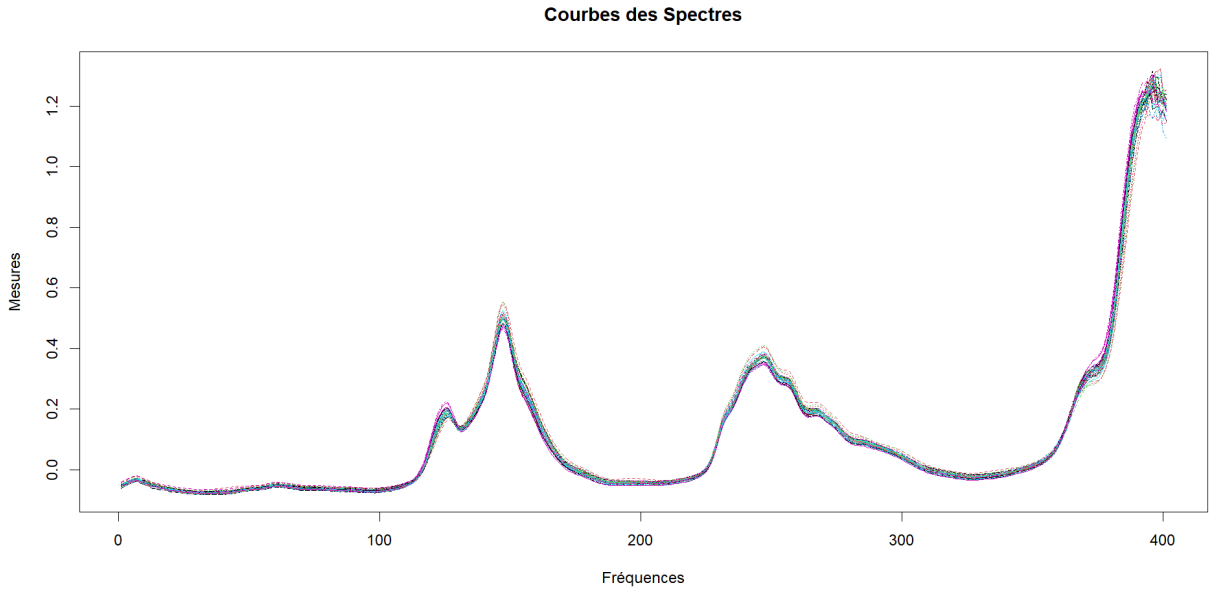
$$\boxed{A_0 = \sum_{j=1}^{r} \frac{1}{\sigma_j} v_j u_i^T}$$

2 Analyse Exploratoire

2.1 Question 1 : Une première approche des données

Nous traçons les boxplots des variables explicatives pour chaque observation du jeu d'apprentissage, et obtenons les tracés suivants :

Aussi, nous nous intéressons aux "courbes" des spectres pour ces mêmes données d'apprentissage. Le tracé obtenu est le suivant :



Ces courbes représentent le profil du spectre infrarouge de chaque échantillon de gasoil. Elles révèlent des zones de fréquences présentant des pics d'intensité, probablement caractéristiques de composants chimiques du gasoil.

Étudier la corrélation entre les mesures aux différentes fréquences fait naturellement appel à la fonction `corrplot` de R. Le nombre de fréquences étant très important, l'exécution du code utilisant cette fonction est long. Ce tracé est donc long à générer, mais également peu explicite. Nous préférerons ainsi visualiser la matrice de corrélation grâce à la fonction `heatmap` (valeurs représentées par des couleurs plus ou moins chaudes). Il résulte de ces fonctions les tracés suivants :

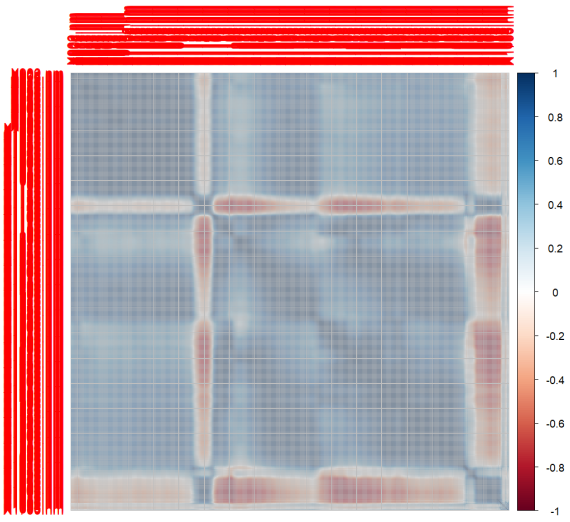


FIGURE 1 – Résultat de la fonction `corrplot`

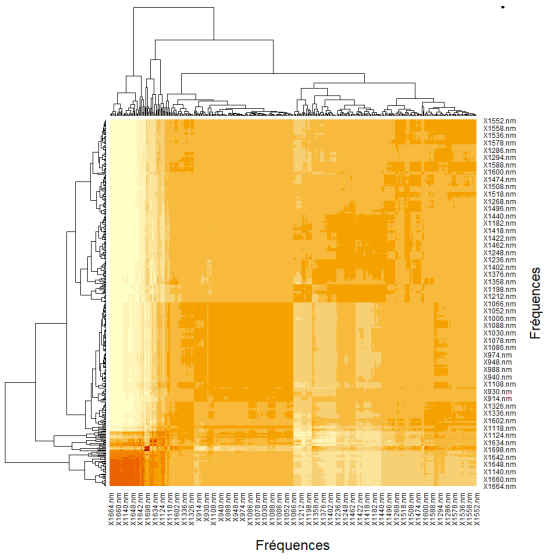
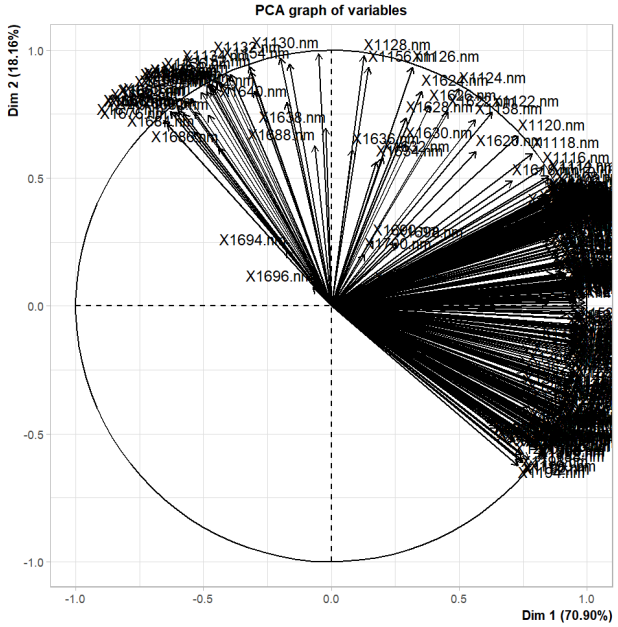


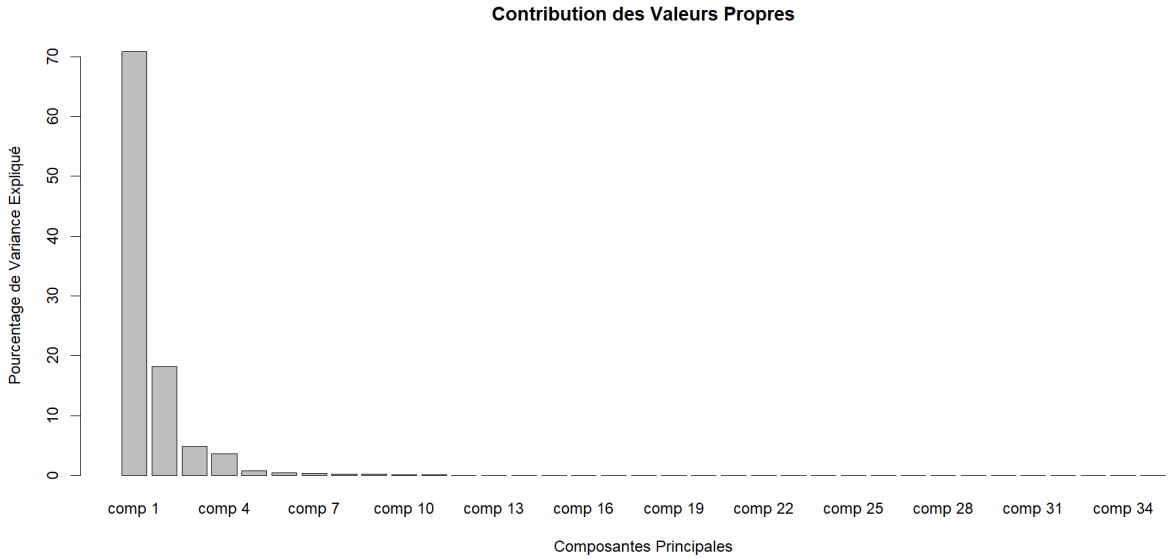
FIGURE 2 – Matrice de corrélation via `heatmap`

2.2 Question 2 : ACP

Nous effectuons une ACP de nos données. Le cercle des corrélations qui en résulte est à la fois long à générer et difficile d'interprétation. Les variables se chevauchent les unes les autres, rendant le cercle fouilli.



Nous traçons le graphe des valeurs propres.



Un coude significatif apparaît entre les cinquième et sixième composantes principales. Afin de nous fixer sur le nombre optimal de composantes principales à conserver, nous nous proposons de visualiser les nuages des individus dans les six premiers axes principaux. Il en résulte que les nuages possèdent encore une inertie non négligeable dans des plans prenant la cinquième dimension pour axe.

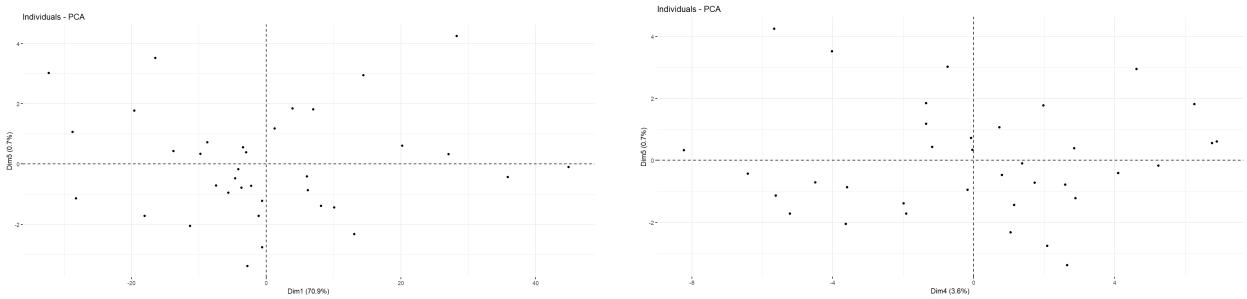


FIGURE 3 – Nuages des individus avec la 5^e composante principale pour axe

En revanche, les nuages prenant la sixième dimension pour axe restent très allongés : leurs inerties en sont tout autant dérisoires.

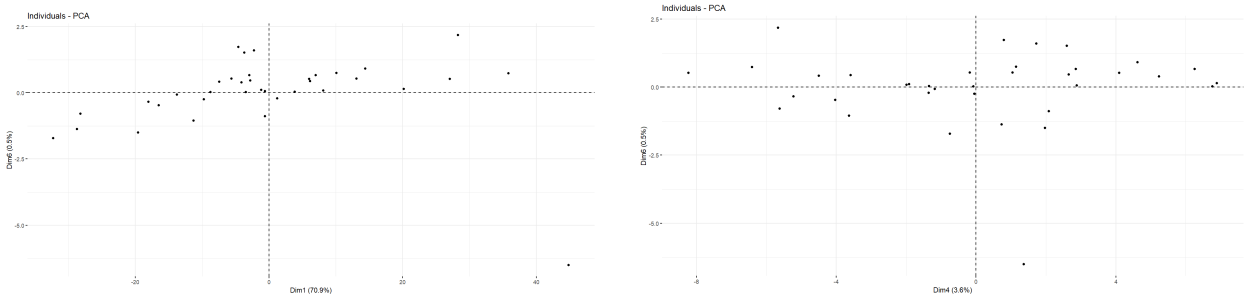


FIGURE 4 – Nuages des individus avec la 6^e composante principale pour axe

Il ne vaut peut-être donc pas la peine de conserver la sixième composante principale. Nous excluons donc l'idée de conserver les composantes principales suivantes.

2.3 Question 3 : Reconstruction du nuage avec l'ACP

La fonction `reconstruct` a été codée pour reconstruire le nuage en utilisant les premiers axes de l'ACP. Voici son implémentation :

```
reconstruct <- function(res, nr, Xm, Xsd) {
  # Sélectionner les premiers nr axes de l'ACP
  axes <- res$ind$coord[,1:nr]
  vars <- t(res$var$coord[,1:nr])
  if(nr!=1){
    eigenvalue_inv <- diag(1/sqrt(res$eig[1:nr,1]))
    reconstructed <- axes %*% eigenvalue_inv %*% vars
  }
  else{
    eigenvalue_inv <- 1/sqrt(res$eig[1:nr,1])
    reconstructed <- eigenvalue_inv * axes %*% vars
  }
  # Réduction des axes (selon écarts-types et moyennes des variables explicatives)
  cloud <- t(t(reconstructed) * Xsd + Xm)
  # Retourner les axes reconstruits
  return(cloud)
}
```

Pour vérifier le code et comparer la reconstruction totale du nuage avec `xtrain`, les erreurs quadratiques moyennes (RMSE) et les erreurs en valeur absolue (MAE) ont été calculées. Ensuite, la reconstruction du nuage a été représentée pour différentes valeurs de `nr` sur une feuille partagée :

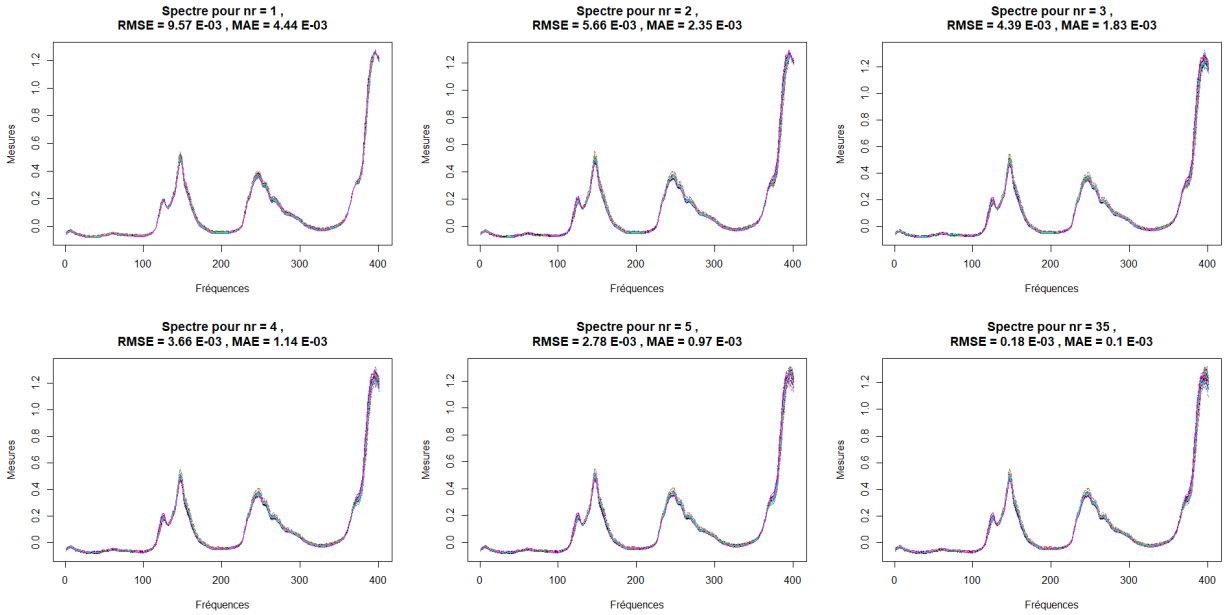


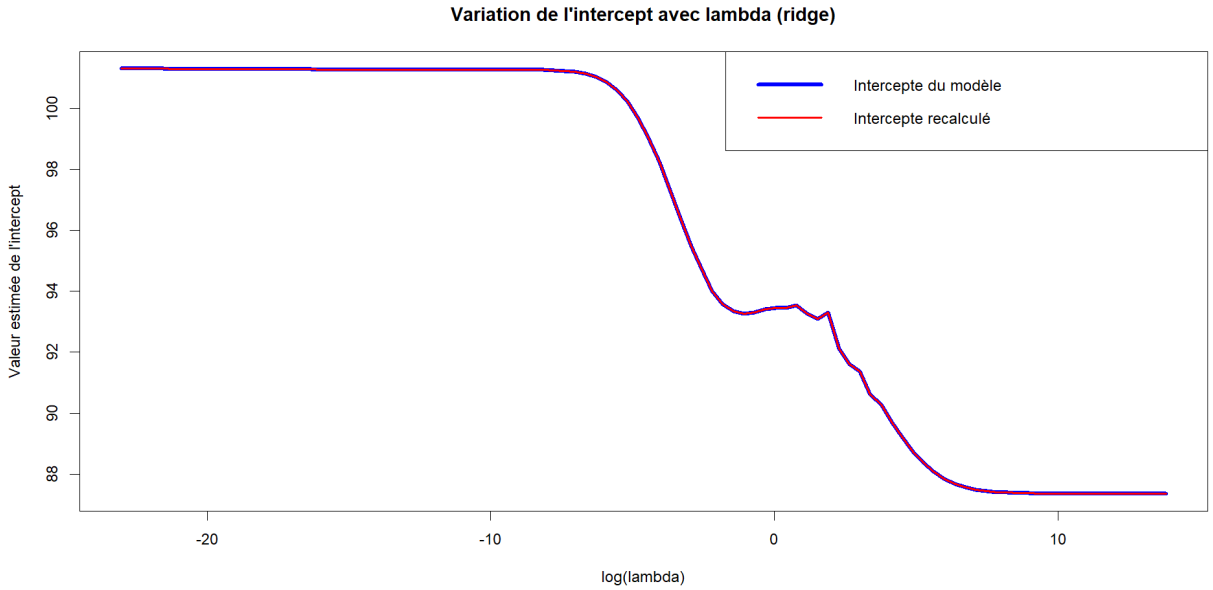
FIGURE 5 – Spectres reconstruits pour différentes valeurs de `nr`

Ces tracés mettent en évidence l'évolution de la qualité de la reconstruction en fonction du nombre de premiers axes utilisés. En effet, les différences sont observables au niveau des pics d'intensité (en particulier le dernier pour lequel la variance des données est grande).

3 Régression pénalisée

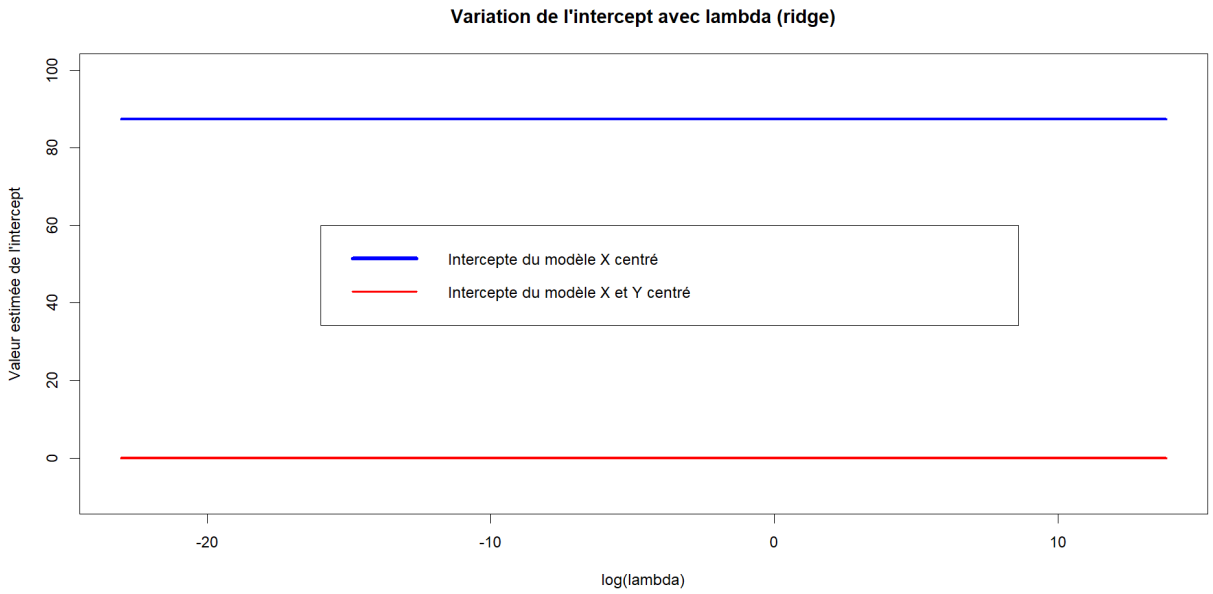
3.1 Question 1 : Un premier modèle de régression ridge

Nous avons estimé le modèle de régression ridge avec la fonction `glmnet` et la grille de paramètres spécifiée. Il en résulte le graphique suivant :



En observant le graphique de la variation de la valeur estimée de l'intercept en fonction de $\log(\lambda)$, nous pouvons visualiser son évolution avec la pénalisation. A mesure que λ augmente, la valeur estimée de l'intercept tend à diminuer. Cette valeur est à peu près constante pour de très petites valeurs et de grandes valeurs de λ.

Ensuite, nous avons recalculé l'intercept en fonction des estimées des autres paramètres. Si nous choisissons de centrer x_{train}, y_{train}, ou les deux, cela affectera la manière dont l'intercept est recalculé.



Centrer x_{train} revient à fixer la valeur de l'intercept à une constante qui semble correspondre à la précédente valeur de stabilisation pour de grandes valeurs de λ. Centrer les deux revient à fixer l'intercept à zéro.

Pour calculer la matrice A_0 dans le cas où y_{train} et les variables de x_{train} sont centrées et réduites, nous utilisons la formule calculée en 1.3 :

$$\boxed{A_0 = \sum_{j=1}^{r} \frac{1}{\sigma_j} v_j u_i^T}$$

Ensuite, pour vérifier que la matrice A_0 est une pseudo-inverse de la matrice centrée réduite du spectre, nous multiplions A_0 par X (matrice centrée réduite du spectre). Si le résultat est proche de l'identité, alors A_0 est une pseudo-inverse correcte.

Pour comparer ce produit à l'identité, on choisit de calculer l'erreur en valeur absolue (MAE). Le calcul donne une valeur de `0.01168504` qui est plutôt proche de zéro : A_0 est donc bien une pseudo-inverse de la matrice centrée réduite du spectre.

Enfin, pour déduire la valeur observée de l'estimateur $\theta_{b\lambda}$ lorsque λ tend vers 0, nous utilisons la formule suivante :

$$\hat{\theta_\lambda} = A_0 \tilde{Y}$$

3.2 Question 2 : Utilisation de la fonction `lm.ridge`

Dans cette section, nous présentons les résultats de l'analyse des données à l'aide de la régression ridge.

Nous avons utilisé la fonction `glmnet` pour estimer le modèle de régression ridge sur les données standardisées. La variation du biais estimé avec lambda est présentée dans la Figure 6.

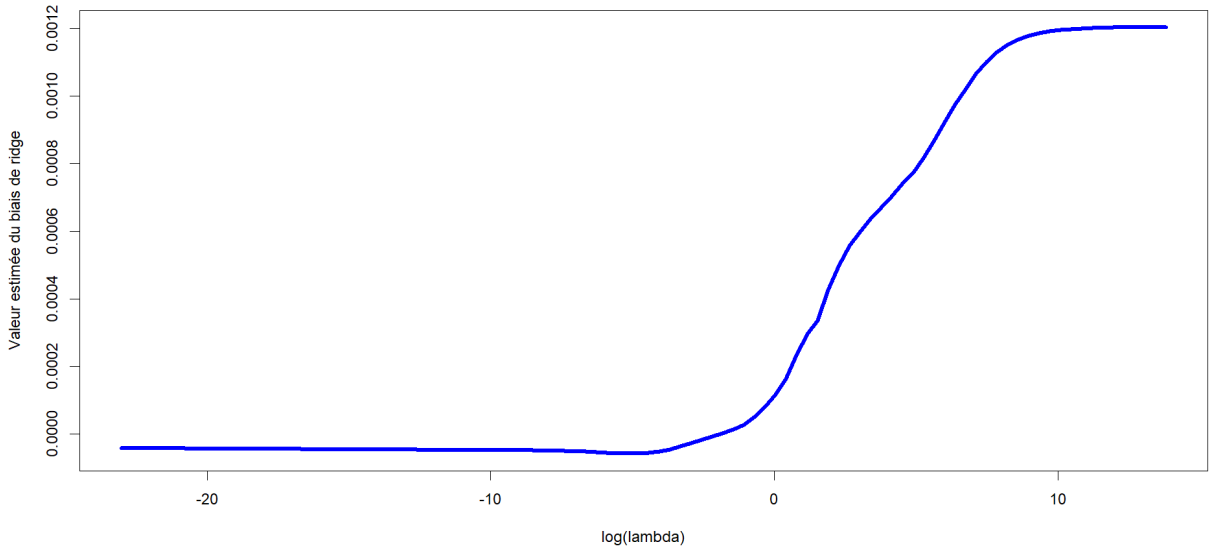


FIGURE 6 – `Variation du biais estimé avec` λ

Nous avons également utilisé la fonction `lm.ridge` pour estimer les coefficients de ridge. La variation du RSE estimé avec lambda est illustrée dans la Figure 12.

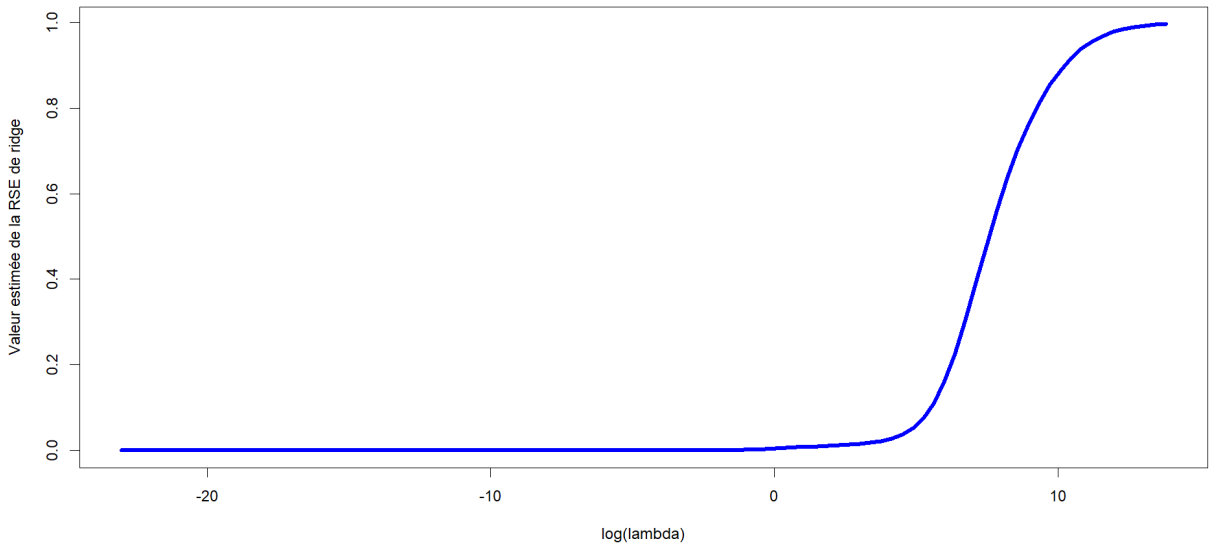


FIGURE 7 – `Variation de la RSE estimée avec` λ

Nous avons calculé les coefficients de ridge directement en utilisant la formule dérivée. La variation de la différence L^2 des coefficients entre les estimations de `glmnet` et les estimations directes est représentée dans la Figure 8.

Nous avons également comparé les résultats de la régression ridge avec une autre famille (gaussienne) pour la régression `glmnet`. La variation de la différence L2 des coefficients pour cette famille est montrée dans la Figure 9.

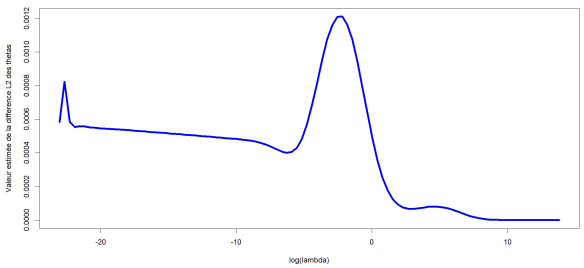


FIGURE 8 – `Variation de la différence` L^2 `des` θ `selon` λ

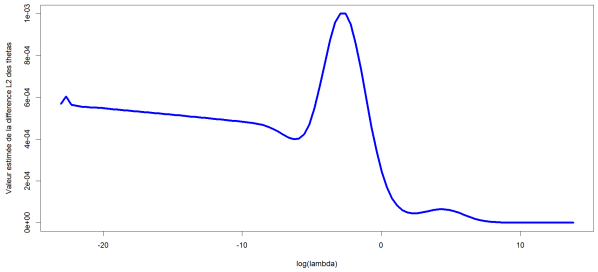


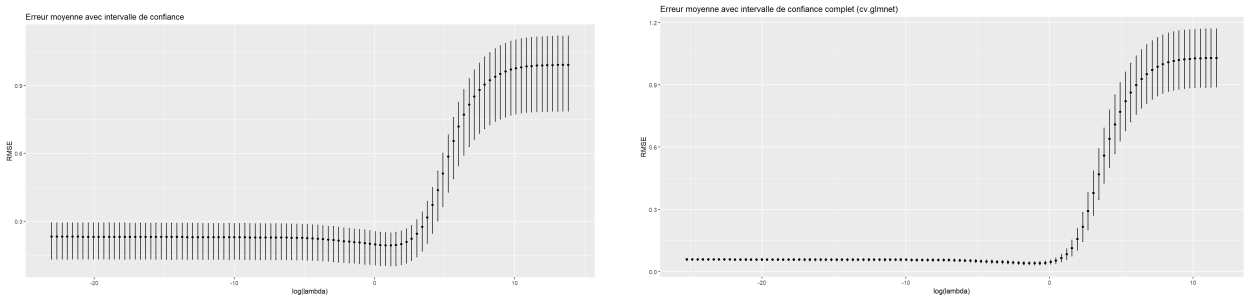
FIGURE 9 – `Différence` L^2 `des` θ `selon` λ `pour une famille gaussienne`

3.3 Question 3

Pour cette question, nous allons d'abord effectuer une validation croisée en quatre plis pour estimer le paramètre de régularisation optimal λ. Ensuite, nous comparerons nos résultats avec ceux de la fonction `cv.glmnet`. Enfin, nous ajusterons le modèle avec le paramètre optimal et calculerons l'erreur de généralisation sur l'échantillon de test.

Validation croisée en quatre plis

Nous avons effectué une validation croisée en quatre plis en utilisant les fonctions `cvsegments` et `glmnet`. Nous avons calculé l'erreur moyenne et l'intervalle de confiance pour chaque valeur du paramètre λ. Nous avons également utilisé la fonction `cv.glmnet` pour comparer nos résultats.



"A la main", grâce à `cv.glmnet` Fonction des auteurs de la librairie

FIGURE 10 – Erreurs moyennes avec intervalles de confiance pour différentes valeurs de λ

La valeur optimale de λ obtenue à partir de notre méthode de validation croisée manuelle est **3.199267**, soit la valeur obtenue par `cv.glmnet`.

Erreur de généralisation

Nous avons entraîné le modèle final avec le paramètre optimal sur l'ensemble du jeu d'apprentissage et avons calculé l'erreur de généralisation sur l'échantillon de test.

— **Erreur de généralisation (RMSE) avec validation croisée manuelle :** 0.16928024137694
— **Erreur de généralisation (RMSE) avec** `cv.glmnet` **:** 0.16928024137694

3.4 Question bonus

La principale différence entre la régression ridge et la régression lasso réside dans la façon dont elles pénalisent les coefficients. La régression ridge utilise une pénalité L^2 (norme Euclidienne), ce qui conduit à des coefficients réduits vers zéro mais pas exactement à zéro. La régression lasso utilise une pénalité L^1 (norme de Manhattan), ce qui favorise la parcimonie en forçant de nombreux coefficients à zéro, ce qui permet la sélection de variables.

Nous avons également comparé les performances de la régression ridge et lasso à l'aide de la fonction `train` avec une validation croisée en cinq plis.

— **Performance du modèle de régression ridge (RMSE) :** 0.553479302878595
— **Performance du modèle de régression lasso (RMSE) :** 0.52576572268207

Cela montre que la régression ridge a légèrement meilleure performance que la régression lasso dans notre cas.

4 Régression pénalisée

4.1 Question 1 : Rappel des hypothèses de la régression logistique et équilibre des jeux de données

Les hypothèses de la régression logistique comprennent :

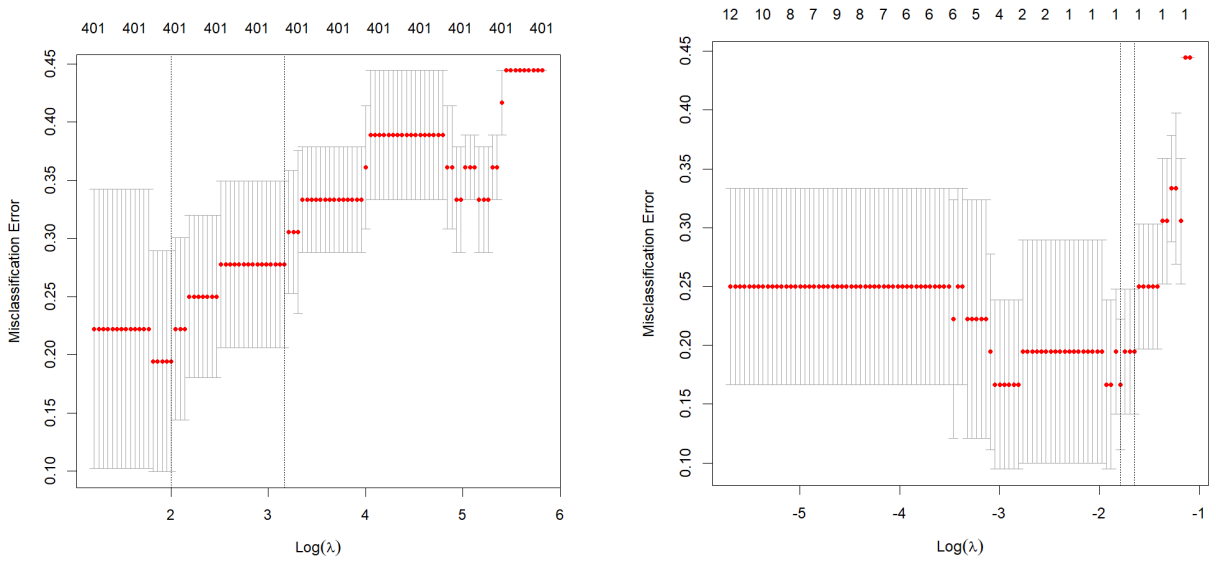
1. **Relation linéaire entre les variables indépendantes et la logit de la variable dépendante** : La régression logistique suppose une relation linéaire entre les variables indépendantes et la logit (logarithme des cotes) de la variable dépendante.
2. **Absence de multicolinéarité** : Les variables indépendantes doivent être peu corrélées entre elles pour éviter la multicollinéarité.
3. **Indépendance des observations** : Chaque observation doit être indépendante des autres. Cela signifie que les observations ne doivent pas être des séries temporelles ou présenter une dépendance structurelle.
4. **Linéarité de la logit pour les variables explicatives** : La logit de la variable dépendante doit être une fonction linéaire des variables explicatives.
5. **Absence d'homoscédasticité et de normalité des résidus** : La variance des résidus ne doit pas être constante à travers les niveaux des variables indépendantes, et les résidus ne doivent pas nécessairement suivre une distribution normale.

Pour créer les variables `z` et `ztest` à prédire, nous avons utilisé la condition où la teneur en octane dépasse strictement le seuil de 88. Nous avons attribué une valeur de 1 à ces observations et une valeur de 0 sinon. Ces variables sont utilisées comme variables dépendantes dans notre modèle de régression logistique.

Nous avons vérifié l'équilibre des jeux d'apprentissage et de test en calculant la proportion d'observations positives (teneur en octane supérieure à 88) par rapport aux observations négatives (teneur en octane inférieure ou égale à 88) dans chaque jeu de données. Les deux jeux de données semblent à peu près équilibrés (rapports de 0.8 et 0.8461538).

4.2 Question 2 : Estimation de la régression logistique pénalisée en ridge et en lasso

Pour estimer la régression logistique pénalisée en ridge et en lasso, nous utilisons la fonction `cv.glmnet` avec $B = 4$ plis pour une validation croisée. L'argument `foldid` est spécifié pour assurer une comparaison équitable entre les deux méthodes. Nous visualisons les résultats de la validation croisée pour les modèles de ridge et de lasso, en affichant les valeurs de l'erreur par pli pour chaque valeur de λ.



Modèle de ridge Modèle de lasso

FIGURE 11 – Erreur de classification en fonction de λ

Nous utilisons les modèles entraînés pour prédire les probabilités d'appartenance à la classe positive pour les données de test. Les probabilités prédites sont ensuite converties en prédictions de classe en utilisant un seuil de 0.5.
Ensuite, nous calculons l'erreur de généralisation pour chaque modèle en comparant les prédictions de classe aux vraies valeurs de la variable de réponse (`ztest`). L'erreur de généralisation est définie comme le taux d'erreur, c'est-à-dire la proportion d'observations mal classées.

Les résultats de l'erreur de généralisation pour la régression logistique pénalisée en ridge et en lasso sont affichés ci-dessous :

— Erreur de généralisation pour la régression logistique pénalisée en ridge : 0.125
— Erreur de généralisation pour la régression logistique pénalisée en lasso : 0.0416666666666667

4.3 Question 3 : Tracé des courbes ROC pour les modèles en ridge et en lasso

Pour comparer les performances des modèles de régression logistique pénalisée en ridge et en lasso, nous utilisons les courbes ROC (*Receiver Operating Characteristic*). Nous utilisons la fonction `roc.glmnet` pour calculer les courbes ROC des modèles retenus en ridge et en lasso. Les courbes ROC sont calculées à la fois sur les données d'apprentissage et sur les données de test, en fournissant les prédictions des modèles sur les données de test et les vraies valeurs de la variable de réponse.

Les courbes ROC ont les tracés suivants :

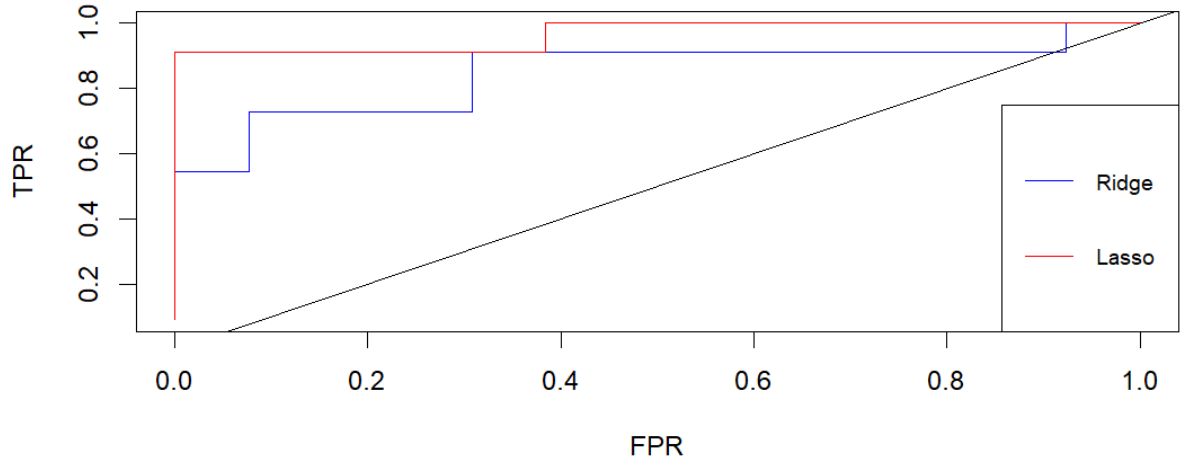


FIGURE 12 – `Courbes ROC des deux modèles`

L'aire sous la courbe (AUC) est une bonne mesure de la performance du modèle. Une AUC plus élevée indique une meilleure performance de classification. Ainsi, la régression lasso a une meilleure performance que la régression ridge dans le cas présent.

5 Un petit bonus ? oh oui !

5.1 Question 1 : C'est beau la symétrie

Pour montrer qu'il existe une famille orthonormée de vecteurs $\{u_j\}$ et des scalaires $\lambda_j > 0$ tels que $XX' = \sum_{j=1}^{r} \lambda_j u_j u_j'$, considérons la matrice XX', qui est symétrique.

Selon le théorème spectral, toute matrice symétrique A admet une décomposition spectrale de la forme :

$$A = Q\Lambda Q'$$

où Q est une matrice orthogonale dont les colonnes sont les vecteurs propres de A, et Λ est une matrice diagonale contenant les valeurs propres correspondantes.

Appliquons ce résultat à la matrice XX'. Puisque XX' est symétrique, il admet une décomposition spectrale :

$$XX' = Q\Lambda Q'$$

où Q est orthogonale et Λ est diagonale.

Ainsi, chaque colonne de Q, que nous notons u_j, est un vecteur propre de XX'. De plus, comme Q est orthogonale, ses colonnes forment une base orthonormée de $\mathbb{R}^p$.

Les éléments diagonaux de Λ, que nous notons λ_j, sont les valeurs propres correspondantes. Puisque XX' est symétrique, ses valeurs propres sont toutes réelles et non négatives.

En résumé, nous avons montré que XX' peut être décomposé comme suit :

$$\boxed{XX' = \sum_{j=1}^{r} \lambda_j u_j u_j'}$$

avec $\{u_j\}$ formant une famille orthonormée de vecteurs propres de XX', $\lambda_j > 0$ étant les valeurs propres correspondantes et $r = \mathbf{rg}(X)$.

5.2 Question 2 : L'image de mon image est mon image

Pour montrer que $M_Q = \sum_{j=1}^{r} u_j u_j'$ est une matrice de projection de $\mathrm{Im}(XX')$, rappelons que les colonnes de Q dans la décomposition spectrale de XX' sont les vecteurs propres de XX'.

Considérons un vecteur x dans $\mathrm{Im}(XX')$. Cela signifie qu'il existe un vecteur y tel que $x = XX'y$. Alors :

$$x = XX'y = Q\Lambda Q'y$$

Puisque Q est orthogonale, $Q'Q = I_n$, donc :

$$Q'x = Q'XX'y = \Lambda Q'y$$

Maintenant, notons $y'Q = \begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_r & \dots & \alpha_n \end{pmatrix}$, où les α_j sont les coefficients de $\alpha_j = < u_j, y >$ (Attention, il est important de remarquer que nous avons des u_j pour $j > r$, ils sont cependant associés a une valeur propre nulle). Alors :

$$Q'x = \Lambda Q'y = \Lambda \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_r \\ \vdots \\ \alpha_n \end{pmatrix} = \begin{pmatrix} \lambda_1\alpha_1 \\ \lambda_2\alpha_2 \\ \vdots \\ \lambda_r\alpha_r \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

$$\text{Donc} \quad x = Q \begin{pmatrix} \lambda_1\alpha_1 \\ \lambda_2\alpha_2 \\ \vdots \\ \lambda_r\alpha_r \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad \text{d'où} \quad x = \sum_{j=1}^{r} \lambda_j \alpha_j u_j$$

Cela nous donne que $M_Q x = \sum_{j=1}^{r} u_j u_j' x = \sum_{i=1}^{r}\sum_{j=1}^{r} \lambda_j\alpha_j u_j u_j' u_i = \sum_{j=1}^{r} \lambda_j\alpha_j u_j = x$ (car les u_j sont orthogonaux donc $u_j' u_i = \delta_{ij}$).

Et donc, $\sum_{j=1}^{r} u_j u_j'$ est une matrice de projection de $\text{Im}(XX')$.

Pour montrer que les projections sur $\text{Im}(X)$ et sur $\text{Im}(XX')$ sont identiques, considérons un vecteur x dans $\text{Im}(X)$. Cela signifie qu'il existe un vecteur y tel que $x = X'y$.

On rappelle que les projections sont des projecteurs et inversement.

Soit p une projection de $\text{Im}(XX')$. Par définition d'un projecteur de $\text{Im}(XX')$: $p^2XX'y = pXX'y$ donc $p^2Q\Lambda Q'Q\Lambda Q'y = pQ\Lambda Q'Q\Lambda Q'y$ donc $p^2Q\Lambda^2Q'y = pQ\Lambda^2Q'y$ et en choisissant correctement y, on atteint tout les vecteurs de $\text{Im}(X)$. Donc les projecteurs de $\text{Im}(XX')$ sont à une déformation près (même dimension r pour les deux images mais pas mêmes éléments) des projecteurs de $\text{Im}(X)$.

Montrons maintenant l'autre sens :

On prend x dans $\text{Im}(XX')$ cette fois-ci. Cela signifie qu'il existe un vecteur y tel que $x = XX'y$. Soit p une projection de $\text{Im}(X)$. On a très vite que $x \in \text{Im}(X)$, en effet $x = X(X'y)$. Donc $p^2x = px$ par définition d'un projecteur, d'où p est un projecteur pour $\text{Im}(XX')$.

On a bien au final que les projections sur $\text{Im}(X)$ et sur $\text{Im}(XX')$ sont identiques.

5.3 Question 3 : Un dénouement singulier

Pour montrer que les v_j sont normés, commençons par calculer leur norme :

$$\|v_j\|^2 = \|\lambda_j^{-1/2} X'u_j\|^2 = (\lambda_j^{-1/2})^2 \|X'u_j\|^2 = \lambda_j^{-1}\|X'u_j\|^2$$

Puisque u_j est un vecteur propre de XX', nous avons $XX'u_j = \lambda_j u_j$. Donc :

$$\lambda_j^{-1}\|X'u_j\|^2 = \lambda_j^{-1} u_j' XX'u_j = \lambda_j^{-1}\lambda_j = 1$$

Ce qui montre que les v_j sont effectivement normés.

Maintenant, montrons que la famille $\{v_j\}$ est une famille orthonormée de vecteurs propres de $X'X$.

Pour montrer l'orthogonalité, prenons deux vecteurs v_i et v_j avec $i \neq j$:

$$v_i' v_j = (\lambda_i^{-1/2} X'u_i)'(\lambda_j^{-1/2} X'u_j) = \lambda_i^{-1/2}\lambda_j^{-1/2} u_i' XX'u_j$$

Puisque u_i est un vecteur propre de XX', nous avons $XX'u_j = \lambda_j u_j$, et donc :

$$v_i' v_j = \lambda_i^{-1/2} \lambda_j^{-1/2} \lambda_j u_i' u_j = 0$$

Ce qui montre que les v_i sont orthogonaux entre eux.
Finalement, montrons que les v_i sont des vecteurs propres de $X'X$. Pour cela, calculons :

$$X'X v_j = X'X(\lambda_j^{-1/2} X' u_j) = \lambda_j^{-1/2} X'XX'u_j = \lambda_j^{-1/2} X'(\lambda_j u_j) = \lambda_j^{-1/2} \lambda_j X' u_j = \lambda_j v_j$$

Ce qui montre que les v_j sont des vecteurs propres de $X'X$ avec des valeurs propres λ_j.
Ainsi, la famille $\{v_j\}$ est une famille orthonormée de vecteurs propres de $X'X$.

Finalement, on remarque que :

$$u_j v_j' = u_j \lambda_j^{-1/2} u_j' X = \lambda_j^{-1/2} u_j u_j' X$$

Donc pour $\sigma_j = \lambda_j^{1/2}$, on a :

$$\sum_{j=1}^{r} \sigma_j u_j v_j' = \sum_{j=1}^{r} u_j u_j' X$$

Or on a démontré que $\sum_{j=1}^{r} u_j u_j'$ était un projecteur de $\mathrm{Im}(XX')$. Et on a démontré que les projecteurs de $\mathrm{Im}(XX')$ étaient des projecteurs de $\mathrm{Im}(X)$. D'où le résultat final tant attendu :

$$\boxed{\sum_{j=1}^{r} \sigma_j u_j v_j' = X}$$

Conclusion

Dans ce rapport, nous avons étudié les techniques de régression pénalisée, en mettant l'accent sur la régression ridge et la régression lasso. Après une analyse initiale des données, incluant des boxplots, des courbes de spectres et une étude de corrélation entre les variables, nous avons réduit la dimensionnalité des données à l'aide d'une analyse en composantes principales (ACP).

De plus, nous avons exploré la décomposition en valeurs singulières comme une alternative à l'ACP pour réduire la dimensionnalité des données et explorer leur structure latente. Cette décomposition offre une représentation compacte des données en décomposant la matrice de données en trois matrices constituées de vecteurs singuliers et de valeurs singulières, permettant une analyse plus approfondie des relations entre les variables.

Ensuite, nous avons appliqué la régression pénalisée en ridge et en lasso, en utilisant la validation croisée pour choisir les paramètres optimaux. Nous avons évalué les performances des modèles en comparant leur erreur de généralisation et en traçant des courbes ROC pour évaluer leur capacité de classification.

Ce projet souligne l'importance des techniques de régression pénalisée dans l'analyse de données, en particulier lorsque le nombre de variables explicatives est élevé par rapport au nombre d'observations, et lorsque la multicollinéarité est présente. Ces techniques offrent un équilibre entre biais et variance, permettant de sélectionner les variables importantes tout en

évitant le surajustement.

Pour pousser l'étude du problème plus loin, une direction intéressante pourrait consister à explorer d'autres méthodes de régression pénalisée, telles que l'élastic net, qui combine à la fois les pénalités de ridge et de lasso ($\alpha \in]0,1[$).

Mot de la fin

Petit mot à l'attention de Mme Keribin :
Nous souhaitions à la fin de ce compte-rendu de projet vous remercier pour votre cours qui a été l'un des cours les plus enrichissant que nous ayons eu jusqu'aujourd'hui à l'ENSTA (et il y a beaucoup de concurrence avec des cours comme MA102, OPT202, AOT13, etc). Nous espérons ne pas avoir dit trop de bêtises dans ce rapport (dans l'espoir de vous rassurer sur ce que les élèves ont pu tirer de votre cours). Merci pour ces séances pleines d'enthousiasme ! Merci pour tout !