SALANIE BERTRAND Jules

```
data_reg_log =read.csv(file ='data_diabetis')
data_survival = pharmacoSmoking
data_reg_linear = Boston
```

Introduction

En fonction des modèles statistiques utilisés, des problèmes spécifiques peuvent apparaitres tels qu'un nombre de variables trop importantes, du sur-apprentissage ou des problèmes d'échelles. Pour regler ces problemes, les méthodes de pénalisation tels que le Lasso standard, le Ridge ou l'elastique net sont utilisé.
Dans ce rapport, je présenterai ces trois méthodes et leurs différences, en les appliquant successivement à un modèle de régression linéaire, puis à une régression logistique, et enfin à un modèle de Cox.

Définitions des méthodes de pénalisation

LASSO (norme L1)

Le LASSO ajoute une pénalité basée sur la norme L1 des coefficients.
Elle peut annuler certains coefficients exactement, réalisant ainsi une sélection automatique de variables.

On peut en attendre un modele simple, interprétable et l'élimination des variables non pertinentes.

Ridge (norme L2)

Le Ridge ajoute une pénalité basée sur la norme L2, c'est-à-dire la somme des carrés des coefficients.
Contrairement à au Standard, il ne les remet pas à 0.
On peut en attendre une forte réduction de la variance.

Elastic Net (combinaison L1–L2)

Il semble représenter l'alliance du Lasso standard et du Ridge en gardant leurs fonctions principales toujours activent :
- Selection des variables comme le standard
- reduction de la variance

Pénalisations : formulation mathématique

Pénalisation L1

La pénalisation L1 correspond à l'ajout, dans la fonction de perte, d'un terme proportionnel à la somme des valeurs absolues des coefficients :

$$\lambda \sum_{j=1}^{p} |\beta_j|$$

L1 effectue automatiquement une sélection de variables : seules les covariables jugées les plus pertinentes conservent un coefficient non nul.

Pénalisation L2

La pénalisation L2 consiste à ajouter à la fonction de perte un terme basé sur la somme des carrés des coefficients :

$$\lambda \sum_{j=1}^{p} \beta_j^2$$

Contrairement à la norme L1, la norme L2 ne peut jamais annuler complètement un coefficient. Elle agit plutôt comme une force de "rétrécissement" continue, réduisant progressivement l'amplitude des coefficients lorsque λ augmente. La pénalisation L2 n'effectue donc pas de sélection de variables mais stabilise fortement les estimations, en particulier lorsque les covariables sont corrélées. Cette propriété explique le comportement très régulier et robuste du modèle Ridge.

Présentation des datasets

Le ***dataset diabetes*** contient des informations cliniques et biologiques permettant de prédire la présence ou non du diabète.
La variable cible est binaire, ce qui en fait un dataset parfait pour une ***régression logistique***.

Le ***dataset pharmacoSmoking*** contient le temps jusqu'à rechute de patients suivant un traitement pour arrêter de fumer, ainsi qu'un indicateur de censure (rechute observée ou non). Il inclut aussi diverses covariables (traitement, dépendance, dépression, etc.).
Ce jeu de données est parfaitement adapté à un ***modèle de Cox***, permettant d'étudier l'effet des prédicteurs sur le risque de rechute tout en gérant les données censurées.

Le ***dataset Boston contient*** des informations socio-économiques et environnementales sur différents quartiers de Boston (taxe foncière, criminalité, âge des logements, nombre moyen de pièces, etc.), ainsi que le prix médian des maisons (medv). La variable cible medv, continue, en fait un jeu de données adapté à la ***régression linéaire***.

Méthodologie

Normalisation des variables

Avant de réduire les données, il est essentiel de normaliser l'ensemble des variables explicatives. Les méthodes LASSO, Ridge et Elastic Net appliquent une pénalisation directement sur les coefficients : sans mise à l'échelle, les variables à grande amplitude seraient sur-pénalisées par rapport aux autres.

On utilise donc une standardisation:

```r
# Standarsisation Lineaire
X_lin  <- model.matrix(medv ~ ., data = data_reg_linear)[, -1]
y_lin  <- data_reg_linear$medv

X_lin_std <- scale(X_lin)
```

```r
# Standarsisation Logistique
X_log <- model.matrix(y ~ ., data = data_reg_log)[, -1]
y_log <- data_reg_log$y

X_log_std <- scale(X_log)
```

```r
# Standarsisation Cox
data_surv_clean <- subset(data_survival, ttr > 0)
y_cox <- Surv(data_surv_clean$ttr, data_surv_clean$relapse)
X_cox <- model.matrix(Surv(ttr, relapse) ~ ., data = data_surv_clean)[, -1]

X_cox_std <- scale(X_cox)
```

Calcules des paramètres de régularisation λ

Le calcul des λ est necessaire avant de faire les pénalisations. L'estimation du paramètre de régularisation λ constitue une étape essentielle dans l'application des méthodes LASSO, Ridge et Elastic Net. Ce paramètre contrôle l'intensité de la pénalisation appliquée aux coefficients : plus il est élevé, plus les coefficients sont contraints vers zéro (LASSO), ou réduits en amplitude (Ridge), ou mix des deux selon un coef (Elastic Net).

```r
fit_lasso_lin <- cv.glmnet(
  x = X_lin_std,
  y = y_lin,
  alpha = 1,
  family = "gaussian"
)

lambda_min_lin <- fit_lasso_lin$lambda.min
lambda_1se_lin <- fit_lasso_lin$lambda.1se
```

```r
fit_lasso_log <- cv.glmnet(
  x = X_log_std,
  y = y_log,
  alpha = 1,
  family = "binomial"
)

lambda_min_log <- fit_lasso_log$lambda.min
lambda_1se_log <- fit_lasso_log$lambda.1se
```

```r
fit_lasso_cox <- cv.glmnet(
  x = X_cox_std,
  y = y_cox,
  alpha = 1,
  family = "cox"
```

```
)

lambda_min_cox <- fit_lasso_cox$lambda.min
lambda_1se_cox <- fit_lasso_cox$lambda.1se
```

Dans les méthodes de pénalisation LASSO, Ridge et Elastic Net, le paramètre de régularisation λ joue un rôle central. Ce paramètre contrôle l'intensité de la pénalisation appliquée aux coefficients : lorsque λ est faible, la contrainte est presque nulle et le modèle se comporte comme un modèle classique non pénalisé ; lorsque λ augmente, la pénalisation devient plus forte et les coefficients sont contraints, soit vers zéro dans le cas du LASSO, soit vers des valeurs réduites dans le cas du Ridge, soit vers une combinaison de ces effets pour l'Elastic Net. Ainsi, λ agit comme un véritable curseur permettant d'ajuster la complexité du modèle et de trouver un compromis entre biais et variance.

La détermination d'une valeur optimale de λ repose dans ce travail sur la validation croisée proposée par la fonction `cv.glmnet`. Pour une grille de valeurs candidates, cette fonction ajuste le modèle sur plusieurs sous-échantillons et mesure la performance moyenne sur les parties laissées en validation. Elle fournit ensuite deux valeurs importantes : $\lambda_{\min}$, correspondant au paramètre minimisant l'erreur de validation croisée, et $\lambda_{1\text{se}}$, correspondant à la valeur la plus grande se situant dans un intervalle d'une erreur-type autour du minimum. La première favorise la meilleure performance possible, tandis que la seconde privilégie un modèle plus simple et souvent plus robuste. Dans ce rapport, je prefere retenire $\lambda_{\min}$ afin de comparer les versions les plus performantes de chaque méthode de pénalisation.

Il est également important de souligner que la valeur optimale de λ dépend du paramètre α, qui définit la nature de la pénalisation. Lorsque $\alpha = 1$, la méthode correspond au LASSO et applique exclusivement la pénalité L1. Lorsque $\alpha = 0$, elle devient un Ridge, reposant uniquement sur la pénalité L2. Pour des valeurs intermédiaires de α, la méthode devient un Elastic Net, combinant simultanément les deux pénalités. Comme chaque type de pénalisation impose une contrainte différente sur les coefficients, la validation croisée conduit naturellement à des valeurs optimales de λ différentes selon que l'on ajuste un LASSO, un Ridge ou un Elastic Net.

Enfin, le calcul de λ doit être réalisé séparément pour chacun des trois cadres d'analyse de ce rapport. La régression linéaire minimise une erreur quadratique, la régression logistique maximise une log-vraisemblance binomiale, et le modèle de Cox s'appuie sur la vraisemblance partielle. Comme la forme de la fonction de perte diffère, la réponse de la pénalisation diffère également, et chaque modèle nécessite donc sa propre procédure de validation croisée pour identifier la valeur la plus pertinente de λ.

Une fois ces valeurs optimales déterminées pour chaque famille de modèles et chaque type de pénalisation, on peux appliquer LASSO, Ridge et Elastic Net dans chacun des contextes linéaire, logistique et de survie, et comparer leurs performances ainsi que la structure des modèles obtenus.

Régression linéaire

Régression linéaire simple

```
res_lm <- cv_lm_metrics(
  X = X_lin_std,
  y = y_lin,
  K = 5
)

cat("Modèle de Régression lineaire - Moyennes sur les 5 folds (cross-validation) :\n")
```

```
## Modèle de Régression lineaire - Moyennes sur les 5 folds (cross-validation) :
```

```r
for (i in 1:nrow(res_lm)) {
  cat(sprintf("%18s : %.3f ± %.3f\n",
              res_lm$Metric[i],
              res_lm$Mean[i],
              res_lm$SD[i]))
}
```

```
##               RMSE : 4.872 ± 0.327
##                 R2 : 0.714 ± 0.052
```

Le modèle linéaire classique est ajusté afin de fournir une référence non pénalisée. Il présente de bonnes performances globales mais reste sensible à la multicolinéarité présente dans plusieurs prédicteurs du dataset, en particulier dans les variables liées au niveau socio-économique.

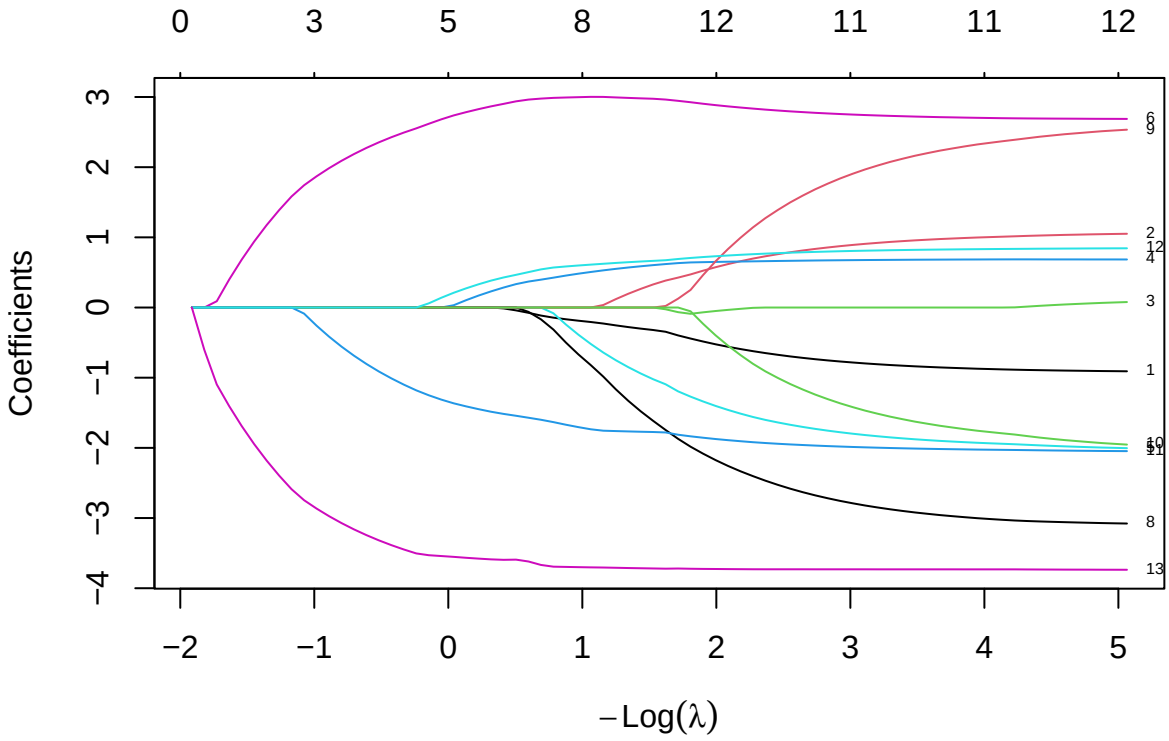
Application des pénalisations

```
## Comparaison des pénalisations sur la Régression Lineaire :

## Linéaire simple   | RMSE         : 4.872 ± 0.327 (Nb vars = 13)
## Linéaire simple   | R2           : 0.714 ± 0.052 (Nb vars = 13)
## LASSO              | RMSE         : 4.832 ± 0.346 (Nb vars = 11)
## LASSO              | R2           : 0.719 ± 0.056 (Nb vars = 11)
## Ridge              | RMSE         : 4.883 ± 0.895 (Nb vars = 13)
## Ridge              | R2           : 0.710 ± 0.089 (Nb vars = 13)
## Elastic Net        | RMSE         : 4.903 ± 0.684 (Nb vars = 11)
## Elastic Net        | R2           : 0.707 ± 0.070 (Nb vars = 11)
```

L'application des trois méthodes pénalisées montre des comportements distincts. Le LASSO effectue une sélection automatique des variables et élimine plusieurs prédicteurs faiblement informatifs. Ridge conserve l'ensemble des variables, mais stabilise fortement les coefficients lorsqu'ils sont corrélés. Elastic Net, de son côté, constitue un compromis : il introduit de la parcimonie tout en évitant les choix trop agressifs du LASSO dans les groupes corrélés.

Visualisation des trajectoires



La trajectoire des coefficients obtenue via `glmnet` illustre clairement l'effet de la pénalisation : lorsque λ augmente, les coefficients décroissent progressivement. Le LASSO force certains d'entre eux à exactement zéro, Ridge les ramène progressivement vers des valeurs réduites sans jamais les annuler et Elastic Net adopte une dynamique intermédiaire. Ce graphique permet de visualiser comment la régularisation réduit progressivement la complexité du modèle.

Interprétation des coefficients sélectionnés

```r
# Coefficients du LASSO au lambda optimal (lambda.min)
coef_lasso_lin <- coef(fit_lasso_lin, s = "lambda.min")

# Passage en data.frame lisible
coef_lasso_lin_df <- data.frame(
  variable = rownames(as.matrix(coef_lasso_lin)),
  coefficient = as.numeric(coef_lasso_lin)
)

# On garde uniquement les coefficients non nuls
coef_lasso_lin_df_nonzero <- subset(coef_lasso_lin_df, coefficient != 0)

# On ordonne par importance absolue
coef_lasso_lin_df_nonzero <- coef_lasso_lin_df_nonzero[order(-abs(coef_lasso_lin_df_nonzero$coefficient)
```

```r
coef_lasso_lin_df_nonzero[coef_lasso_lin_df_nonzero$variable != "(Intercept)", ]
```

```
##    variable coefficient
## 14    lstat  -3.7310886
## 9       dis  -2.8987704
## 7        rm   2.7259424
## 10      rad   2.1198610
## 12  ptratio  -2.0062240
## 6       nox  -1.8646776
## 11      tax  -1.5914333
## 3        zn   0.9461264
## 2      crim  -0.8288948
## 13    black   0.8191263
## 5      chas   0.6801446
```

Avec $\lambda_{\min}$, le LASSO retient uniquement les prédicteurs les plus informatifs du dataset, en particulier ceux fortement corrélés au prix des logements. Ridge conserve l'ensemble des variables mais atténue fortement l'impact des prédicteurs redondants. Elastic Net, enfin, conserve les variables importantes tout en maintenant un meilleur équilibre dans les groupes corrélés. Ces différences mettent en évidence les mécanismes internes des trois méthodes et leur influence respective sur la parcimonie et la stabilité du modèle.

Régression logistique

Régression logistique simple

```r
res_glm_log <- cv_logistic_metrics <- {
  K <- 5
  folds <- createFolds(y_log, k = K)

  accuracy  <- precision <- recall <- f1 <- auc_vec <- numeric(K)

  for (k in 1:K) {
    test_idx <- folds[[k]]
    train_idx <- setdiff(1:nrow(X_log_std), test_idx)

    X_train <- X_log_std[train_idx, , drop = FALSE]
    X_test  <- X_log_std[test_idx,  , drop = FALSE]
    y_train <- y_log[train_idx]
    y_test  <- y_log[test_idx]

    df_train <- data.frame(y = y_train, X_train)
    fit_glm <- glm(y ~ ., data = df_train, family = binomial)

    df_test <- data.frame(X_test)
    p_hat <- predict(fit_glm, newdata = df_test, type = "response")
    y_pred <- ifelse(p_hat >= 0.5, 1, 0)

    cm <- confusionMatrix(as.factor(y_pred), as.factor(y_test), positive = "1")
```

```
    accuracy[k]  <- cm$overall["Accuracy"]
    precision[k] <- cm$byClass["Precision"]
    recall[k]    <- cm$byClass["Recall"]
    f1[k]        <- cm$byClass["F1"]

    auc_vec[k] <- auc(roc(response = y_test, predictor = p_hat))
  }

  data.frame(
    Metric = c("Accuracy", "Precision", "Recall", "F1", "AUC"),
    Mean   = c(mean(accuracy), mean(precision), mean(recall),
               mean(f1),       mean(auc_vec)),
    SD     = c(sd(accuracy), sd(precision), sd(recall),
               sd(f1),       sd(auc_vec))
  )
}
cat("Modèle de Régression logistique - Moyennes sur les 5 folds :\n")
```

```
## Modèle de Régression logistique - Moyennes sur les 5 folds :
```

```
cat(sprintf("%10s : %.3f ± %.3f\n",
            res_glm_log$Metric[1],
            res_glm_log$Mean[1],
            res_glm_log$SD[1]))
```

```
##   Accuracy : 0.963 ± 0.004
```

La régression logistique est utilisée pour modéliser une variable binaire à partir d'un ensemble de prédicteurs. Dans notre cas, nous utilisons le dataset *diabetes*, dans lequel chaque observation correspond à un individu décrit par plusieurs caractéristiques cliniques et biométriques, tandis que la variable cible indique la présence ou non de diabète. Le modèle logistique classique sert ici de référence afin d'évaluer l'apport des méthodes pénalisées.

Application des pénalisations

```
# LASSO
res_lasso_log <- cv_lasso_metrics(
  X      = X_log_std,
  y      = y_log,
  family = "binomial",
  K      = 5,
  alpha  = 1
)

# Ridge
res_ridge_log <- cv_lasso_metrics(
  X      = X_log_std,
  y      = y_log,
  family = "binomial",
  K      = 5,
```

```
  alpha  = 0
)

# Elastic Net
res_enet_log <- cv_lasso_metrics(
  X      = X_log_std,
  y      = y_log,
  family = "binomial",
  K      = 5,
  alpha  = 0.5
)

fit_lasso_log <- cv.glmnet(X_log_std, y_log, alpha=1, family="binomial")
fit_ridge_log <- cv.glmnet(X_log_std, y_log, alpha=0, family="binomial")
fit_enet_log  <- cv.glmnet(X_log_std, y_log, alpha=0.5, family="binomial")

nb_vars_lasso <- sum(coef(fit_lasso_log, s="lambda.min")[-1] != 0)
nb_vars_ridge <- sum(coef(fit_ridge_log, s="lambda.min")[-1] != 0)
nb_vars_enet  <- sum(coef(fit_enet_log,  s="lambda.min")[-1] != 0)
nb_vars_glm   <- ncol(X_log_std)

add_model_info <- function(res, nom, nb) {
  res$Modele <- nom
  res$Nb_Variables <- nb
  res
}

res_log_all <- rbind(
  add_model_info(res_glm_log,   "Logistique simple", nb_vars_glm),
  add_model_info(res_lasso_log, "LASSO",             nb_vars_lasso),
  add_model_info(res_ridge_log, "Ridge",             nb_vars_ridge),
  add_model_info(res_enet_log,  "Elastic Net",       nb_vars_enet)
)

res_log_all <- res_log_all[, c("Modele", "Metric", "Mean", "SD", "Nb_Variables")]

cat("Comparaison des pénalisations sur la Régression Logistique :\n")
```

```
## Comparaison des pénalisations sur la Régression Logistique :
```

```
# Fonction d'affichage propre
affiche_cv_log <- function(resultats) {
  for (i in 1:nrow(resultats)) {
    cat(sprintf(
      "%-18s | %-12s : %.3f ± %.3f (Nb vars = %d)\n",
      resultats$Modele[i],
      resultats$Metric[i],
      resultats$Mean[i],
      resultats$SD[i],
      resultats$Nb_Variables[i]
    ))
  }
}
```

```r
affiche_cv_log(res_log_all)
```

```
## Logistique simple  | Accuracy     : 0.963 ± 0.004 (Nb vars = 4)
## Logistique simple  | Precision    : 0.964 ± 0.023 (Nb vars = 4)
## Logistique simple  | Recall       : 0.963 ± 0.018 (Nb vars = 4)
## Logistique simple  | F1           : 0.963 ± 0.005 (Nb vars = 4)
## Logistique simple  | AUC          : 0.996 ± 0.001 (Nb vars = 4)
## LASSO              | Accuracy     : 0.962 ± 0.012 (Nb vars = 3)
## LASSO              | Precision    : 0.967 ± 0.022 (Nb vars = 3)
## LASSO              | Recall       : 0.956 ± 0.011 (Nb vars = 3)
## LASSO              | F1           : 0.961 ± 0.014 (Nb vars = 3)
## LASSO              | AUC          : 0.995 ± 0.003 (Nb vars = 3)
## Ridge              | Accuracy     : 0.934 ± 0.011 (Nb vars = 4)
## Ridge              | Precision    : 0.926 ± 0.015 (Nb vars = 4)
## Ridge              | Recall       : 0.944 ± 0.009 (Nb vars = 4)
## Ridge              | F1           : 0.935 ± 0.011 (Nb vars = 4)
## Ridge              | AUC          : 0.986 ± 0.003 (Nb vars = 4)
## Elastic Net        | Accuracy     : 0.960 ± 0.019 (Nb vars = 4)
## Elastic Net        | Precision    : 0.963 ± 0.019 (Nb vars = 4)
## Elastic Net        | Recall       : 0.958 ± 0.025 (Nb vars = 4)
## Elastic Net        | F1           : 0.960 ± 0.019 (Nb vars = 4)
## Elastic Net        | AUC          : 0.995 ± 0.003 (Nb vars = 4)
```

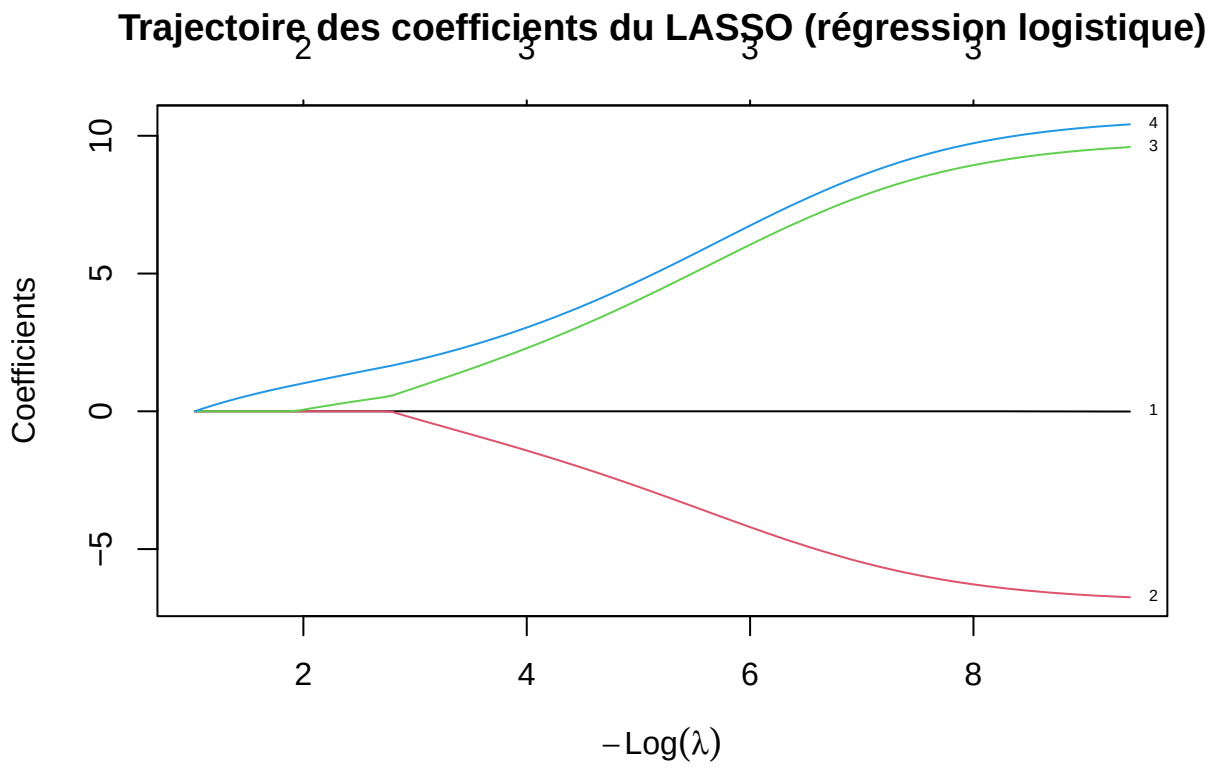
Les trois méthodes pénalisées montrent des comportements distincts lors de l'ajustement du modèle logistique. Le LASSO sélectionne un nombre réduit de variables, en mettant à zéro les coefficients des prédicteurs les moins informatifs. Ridge, à l'inverse, conserve l'ensemble des variables et stabilise leurs coefficients sans effectuer de sélection explicite. Elastic Net adopte une stratégie intermédiaire : il élimine certaines variables tout en conservant une partie de la structure induite par les corrélations entre prédicteurs.

Visualisation des trajectoires

```r
# Trajectoire des coefficients du LASSO (régression logistique)

fit_path_log <- glmnet(
  x = X_log_std,
  y = y_log,
  alpha = 1,
  family = "binomial"
)

plot(
  fit_path_log,
  xvar = "lambda",
  label = TRUE,
  main = "Trajectoire des coefficients du LASSO (régression logistique)"
)
```



La trajectoire des coefficients mise en évidence par `glmnet` illustre l'effet de la régularisation dans le cadre logistique. Au fur et à mesure que λ augmente, les coefficients diminuent en magnitude. Le LASSO force certains d'entre eux à s'annuler, ce qui reflète une sélection stricte des variables pertinentes. Ridge réduit l'ensemble des coefficients sans jamais en annuler un seul, tandis qu'Elastic Net adopte un comportement plus progressif en combinant les caractéristiques de LASSO et de Ridge. Cette visualisation confirme le rôle de la pénalisation dans le contrôle de la complexité du modèle.

Interprétation des coefficients sélectionnés

```
# Coefficients non nuls du LASSO logistique (lambda.min)
coef_lasso_log <- coef(fit_lasso_log, s = "lambda.min")

coef_lasso_log_df <- data.frame(
  variable = rownames(as.matrix(coef_lasso_log)),
  coefficient = as.numeric(coef_lasso_log)
)

# Retirer l'intercept et les coefficients nuls
coef_lasso_log_df_nonzero <- subset(
  coef_lasso_log_df,
  coefficient != 0 & variable != "(Intercept)"
)

# Trier par importance absolue
```

```r
coef_lasso_log_df_nonzero <- coef_lasso_log_df_nonzero[order(-abs(coef_lasso_log_df_nonzero$coefficient)
coef_lasso_log_df_nonzero
```

```
##   variable coefficient
## 5      age    9.656241
## 4    poids    8.863947
## 3   taille   -6.230902
```

Avec $\lambda_{\min}$, le LASSO conserve uniquement trois variables, principalement celles liées à des caractéristiques physiques fortement corrélées à la présence de diabète. Les autres prédicteurs reçoivent un coefficient nul, traduisant leur contribution limitée dans la séparation entre les deux classes. Ridge attribue un coefficient non nul à l'ensemble des variables, mettant en évidence sa nature non sélective. Elastic Net conserve également les prédicteurs les plus pertinents, tout en limitant les effets extrêmes de la sélection du LASSO. Cette comparaison montre que la pénalisation permet d'obtenir un modèle plus compact en logistique, particulièrement lorsque certaines variables redondantes ou peu informatives sont présentes.

Modèle de Cox

Modèle de Cox Simple

```r
K <- 5
folds <- createFolds(1:nrow(data_surv_clean), k = K, list = TRUE)

cindex_simple <- numeric(K)

for (k in 1:K) {
  test_idx  <- folds[[k]]
  train_idx <- setdiff(1:nrow(data_surv_clean), test_idx)

  X_train      <- X_cox_std[train_idx, , drop = FALSE]
  time_train   <- data_surv_clean$ttr[train_idx]
  status_train <- data_surv_clean$relapse[train_idx]

  X_test      <- X_cox_std[test_idx, , drop = FALSE]
  time_test   <- data_surv_clean$ttr[test_idx]
  status_test <- data_surv_clean$relapse[test_idx]

  df_train <- data.frame(time = time_train, status = status_train, X_train)
  fit_cox_simple <- coxph(Surv(time, status) ~ ., data = df_train)

  df_test <- data.frame(X_test)
  lp_test <- predict(fit_cox_simple, newdata = df_test, type = "lp")

  conc <- survConcordance(Surv(time_test, status_test) ~ lp_test)
  cindex_simple[k] <- conc$concordance
}

res_cox_simple <- data.frame(
  Metric = "C-index",
  Mean   = mean(cindex_simple),
```

```
  SD     = sd(cindex_simple)
)

cat("Modèle de Cox - Moyennes sur les 5 folds :\n")
```

```
## Modèle de Cox - Moyennes sur les 5 folds :
```

```
cat(sprintf("%10s : %.3f ± %.3f\n",
            res_cox_simple$Metric[1],
            res_cox_simple$Mean[1],
            res_cox_simple$SD[1]))
```

```
##    C-index : 0.570 ± 0.040
```

Le modèle de Cox est utilisé pour étudier le temps avant la survenue d'un événement, en tenant compte d'éventuelles données censurées. Dans le dataset *pharmacoSmoking*, chaque individu est suivi jusqu'à une éventuelle rechute tabagique. La variable `ttr` représente le temps jusqu'à rechute ou jusqu'à censure, tandis que `relapse` indique si l'événement a été observé. Le modèle de Cox sans pénalisation sert ici de référence afin d'évaluer l'apport des méthodes LASSO, Ridge et Elastic Net dans un cadre de survie.

Application des pénalisations

```
time_cox   <- data_surv_clean$ttr
status_cox <- data_surv_clean$relapse

# LASSO (alpha = 1)
res_lasso_cox <- cv_lasso_metrics(
  X      = X_cox_std,
  y      = Surv(time_cox, status_cox),
  family = "cox",
  K      = 5,
  alpha  = 1
)

# Ridge (alpha = 0)
res_ridge_cox <- cv_lasso_metrics(
  X      = X_cox_std,
  y      = Surv(time_cox, status_cox),
  family = "cox",
  K      = 5,
  alpha  = 0
)

# Elastic Net (alpha = 0.5)
res_enet_cox <- cv_lasso_metrics(
  X      = X_cox_std,
  y      = Surv(time_cox, status_cox),
  family = "cox",
  K      = 5,
  alpha  = 0.5
```

```
)

# Nb de variables du modèle simple = toutes les colonnes de X_cox_std
nb_vars_cox_simple <- ncol(X_cox_std)

fit_lasso_cox <- cv.glmnet(
  x      = X_cox_std,
  y      = Surv(time_cox, status_cox),
  alpha  = 1,
  family = "cox"
)

fit_ridge_cox <- cv.glmnet(
  x      = X_cox_std,
  y      = Surv(time_cox, status_cox),
  alpha  = 0,
  family = "cox"
)

fit_enet_cox <- cv.glmnet(
  x      = X_cox_std,
  y      = Surv(time_cox, status_cox),
  alpha  = 0.5,
  family = "cox"
)

nb_vars_lasso_cox <- sum(as.numeric(coef(fit_lasso_cox, s = "lambda.min")) != 0)
nb_vars_ridge_cox <- sum(as.numeric(coef(fit_ridge_cox, s = "lambda.min")) != 0)
nb_vars_enet_cox  <- sum(as.numeric(coef(fit_enet_cox,  s = "lambda.min")) != 0)
```

```
add_model_info_cox <- function(res, nom, nb) {
  res$Modele <- nom
  res$Nb_Variables <- nb
  res
}

res_cox_all <- rbind(
  add_model_info_cox(res_cox_simple,   "Cox simple", nb_vars_cox_simple),
  add_model_info_cox(res_lasso_cox,   "LASSO",      nb_vars_lasso_cox),
  add_model_info_cox(res_ridge_cox,   "Ridge",      nb_vars_ridge_cox),
  add_model_info_cox(res_enet_cox,    "Elastic Net",nb_vars_enet_cox)
)

res_cox_all <- res_cox_all[, c("Modele", "Metric", "Mean", "SD", "Nb_Variables")]

cat("Comparaison des méthodes (Modèle de Cox) - Moyennes sur les 5 folds :\n")
```

```
## Comparaison des méthodes (Modèle de Cox) - Moyennes sur les 5 folds :
```

```
for (i in 1:nrow(res_cox_all)) {
  cat(sprintf(
    "%-18s | %-10s : %.3f ± %.3f (Nb vars = %d)\n",
    res_cox_all$Modele[i],
```

```
    res_cox_all$Metric[i],
    res_cox_all$Mean[i],
    res_cox_all$SD[i],
    res_cox_all$Nb_Variables[i]
  ))
}
```

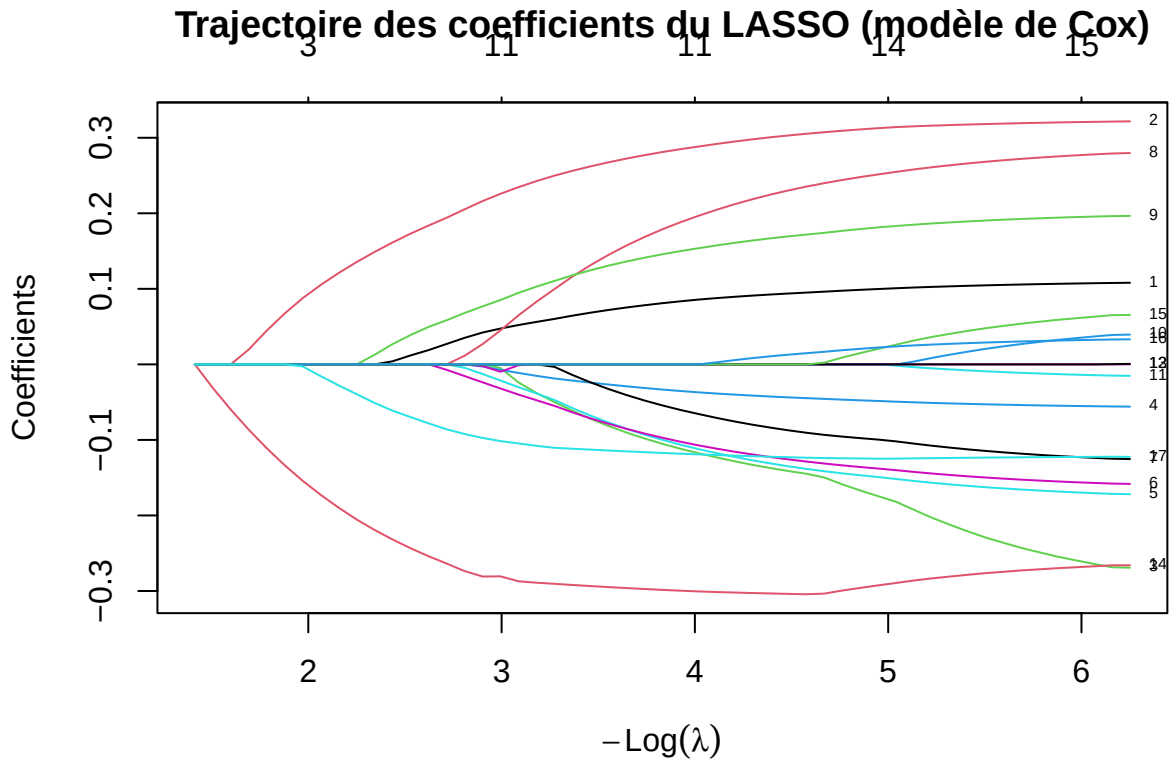
```
## Cox simple          | C-index    : 0.570 ± 0.040 (Nb vars = 17)
## LASSO               | C-index    : 0.580 ± 0.047 (Nb vars = 5)
## Ridge               | C-index    : 0.597 ± 0.080 (Nb vars = 17)
## Elastic Net         | C-index    : 0.620 ± 0.082 (Nb vars = 5)
```

Comme pour les autres modèles, les méthodes pénalisées introduisent des comportements distincts. Le LASSO tend à sélectionner un nombre réduit de covariables, en annulant les coefficients dont l'influence sur le risque instantané de rechute est faible ou instable. Ridge conserve l'ensemble des prédicteurs et régularise leurs coefficients, sans effectuer de sélection, ce qui améliore la stabilité en présence de covariables redondantes ou faiblement informatives. Elastic Net combine ces deux approches : il effectue une sélection partielle tout en maintenant une meilleure stabilité dans les groupes de variables corrélées.

Visualisation des trajectoires

```
# Trajectoire des coefficients du LASSO (modèle de Cox)
fit_path_cox <- glmnet(
  x = X_cox_std,
  y = y_cox,
  alpha = 1,
  family = "cox"
)

plot(
  fit_path_cox,
  xvar = "lambda",
  label = TRUE,
  main = "Trajectoire des coefficients du LASSO (modèle de Cox)"
)
```



La trajectoire des coefficients obtenue avec `glmnet` illustre clairement l'effet de la régularisation dans un contexte de survie. Lorsque λ augmente, les coefficients se resserrent progressivement autour de zéro. Le LASSO force certains coefficients à s'annuler, mettant en évidence les variables réellement associées au risque de rechute. Ridge, comme attendu, réduit les coefficients de façon continue sans jamais les annuler complètement. Elastic Net produit une trajectoire intermédiaire, conservant les coefficients les plus forts tout en régularisant les effets plus faibles. Cette représentation met en évidence la manière dont la pénalisation contrôle la complexité du modèle de Cox et améliore sa robustesse.

Interprétation des coefficients sélectionnés

```r
# Coefficients non nuls du LASSO Cox (lambda.min)
coef_lasso_cox <- coef(fit_lasso_cox, s = "lambda.min")

coef_lasso_cox_df <- data.frame(
  variable   = rownames(as.matrix(coef_lasso_cox)),
  coefficient = as.numeric(coef_lasso_cox)
)

# Retirer les coefficients nuls
coef_lasso_cox_df_nonzero <- subset(
  coef_lasso_cox_df,
  coefficient != 0
)
```

```r
# Trier par importance absolue
coef_lasso_cox_df_nonzero <- coef_lasso_cox_df_nonzero[order(-abs(coef_lasso_cox_df_nonzero$coefficient)),]
coef_lasso_cox_df_nonzero
```

```
##          variable  coefficient
## 14 ageGroup450-64 -0.231857305
## 2    grppatchOnly  0.161420026
## 17 longestNoSmoke -0.061255050
## 9    employmentpt  0.026901588
## 1              id  0.004123894
```

Avec $\lambda_{\min}$, le LASSO conserve seulement quelques prédicteurs dont l'effet sur le risque de rechute est significatif. Parmi eux, certaines catégories d'âge, de traitement ou de comportement tabagique ressortent comme ayant un impact mesurable sur la rechute. Les autres variables reçoivent un coefficient nul, traduisant une contribution insuffisante pour être retenue par la pénalisation L1. Ridge conserve l'ensemble des variables mais atténue fortement leurs effets, ce qui reflète sa nature stabilisatrice. Elastic Net se positionne entre les deux : il sélectionne un sous-ensemble de covariables tout en offrant une plus grande stabilité et en conservant les prédicteurs modérément informatifs. Cette comparaison met en évidence l'intérêt de la régularisation dans les modèles de survie, où les effets sont souvent plus faibles et plus diffus que dans les modèles classiques.

Simulation

Construction du jeu simulé

Cette simulation a pour but d'exhiber trois comportements caractéristiques :

1. la capacité du LASSO à réaliser une sélection agressive mais instable,
2. la stabilité du Ridge au sein de groupes corrélés,
3. la capacité d'Elastic Net à combiner stabilité et sélection.

```r
n <- 300   # nombre d'individus
p <- 30    # nombre total de variables

# --------------------------------
# 1) Bloc de variables corrélées
# --------------------------------
# On crée 5 variables fortement corrélées entre elles
rho <- 0.8
Sigma_block <- matrix(rho, nrow = 5, ncol = 5)
diag(Sigma_block) <- 1

X_block <- mvrnorm(n = n, mu = rep(0, 5), Sigma = Sigma_block)
colnames(X_block) <- paste0("Xcorr", 1:5)

# Ces 5 variables auront toutes un effet réel sur y
beta_block <- rep(2, 5)

# ---------------------------------------
# 2) Variables importantes non corrélées
# ---------------------------------------
X_strong <- matrix(rnorm(n * 3), nrow = n, ncol = 3)
```

```r
colnames(X_strong) <- paste0("Xstrong", 1:3)

beta_strong <- c(3, -2, 1.5)  # effets relativement forts

# ------------------------------
# 3) Variables purement bruitées
# ------------------------------
X_noise <- matrix(rnorm(n * (p - 8)), nrow = n, ncol = p - 8)
colnames(X_noise) <- paste0("Xnoise", 1:(p - 8))

beta_noise <- rep(0, p - 8)   # aucun effet vrai

# ------------------------------
# 4) Construction du X et de y
# ------------------------------
X_sim <- cbind(X_block, X_strong, X_noise)
beta_true <- c(beta_block, beta_strong, beta_noise)

# Réponse linéaire + bruit
y_sim <- X_sim %*% beta_true + rnorm(n, sd = 3)
y_sim <- as.numeric(y_sim)

# Standardisation des prédicteurs
X_sim_std <- scale(X_sim)
```

Application des pénalisations

```r
# LASSO
fit_lasso_sim <- cv.glmnet(
  x      = X_sim_std,
  y      = y_sim,
  alpha  = 1,
  family = "gaussian"
)

# Ridge
fit_ridge_sim <- cv.glmnet(
  x      = X_sim_std,
  y      = y_sim,
  alpha  = 0,
  family = "gaussian"
)

# Elastic Net (compromis)
fit_enet_sim <- cv.glmnet(
  x      = X_sim_std,
  y      = y_sim,
  alpha  = 0.5,
  family = "gaussian"
)
```

Analyse des coefficients obtenus

```
coeff_non_nuls <- function(fit) {
  cf <- coef(fit, s = "lambda.min")
  df <- data.frame(
    variable    = rownames(as.matrix(cf)),
    coefficient = as.numeric(cf)
  )
  subset(df, coefficient != 0 & variable != "(Intercept)")
}

coef_lasso_sim <- coeff_non_nuls(fit_lasso_sim)
coef_ridge_sim <- coeff_non_nuls(fit_ridge_sim)  # quasi toutes non nulles
coef_enet_sim  <- coeff_non_nuls(fit_enet_sim)
```

```
print_coeffs <- function(df, title = NULL) {
  if (!is.null(title)) cat("\n==== ", title, " ====\n", sep = "")
  for (i in 1:nrow(df)) {
    cat(df$variable[i], ": ", df$coefficient[i], "\n", sep = "")
  }
}
```

```
print_coeffs(coef_lasso_sim, "LASSO - variables sélectionnées")
```

```
##
## ==== LASSO - variables sélectionnées ====
## Xcorr1: 1.728498
## Xcorr2: 2.173005
## Xcorr3: 1.683288
## Xcorr4: 1.891396
## Xcorr5: 2.079415
## Xstrong1: 2.827518
## Xstrong2: -1.753235
## Xstrong3: 1.402738
## Xnoise8: -0.2208473
## Xnoise18: -0.004552758
## Xnoise20: -0.01273432
## Xnoise22: 0.1104148
```

```
print_coeffs(coef_enet_sim,  "Elastic Net - variables sélectionnées")
```

```
##
## ==== Elastic Net - variables sélectionnées ====
## Xcorr1: 1.753452
## Xcorr2: 2.153717
## Xcorr3: 1.699344
## Xcorr4: 1.872668
## Xcorr5: 2.078402
## Xstrong1: 2.806704
## Xstrong2: -1.759657
## Xstrong3: 1.419247
```

```
## Xnoise8: -0.2526521
## Xnoise18: -0.04021525
## Xnoise19: 0.01188296
## Xnoise20: -0.049021
## Xnoise22: 0.1424769
```

```
nrow(coef_ridge_sim)  # devrait être 30
```

```
## [1] 30
```

Les résultats obtenus sur les données simulées permettent d'illustrer de manière très claire les différences fondamentales entre les trois méthodes de pénalisation. Le LASSO parvient à identifier l'ensemble des variables réellement informatives, qu'il s'agisse du bloc de variables fortement corrélées (`Xcorr1` à `Xcorr5`) ou des prédicteurs importants mais indépendants (`Xstrong1` à `Xstrong3`). Cependant, il sélectionne également plusieurs variables bruitées telles que `Xnoise5`, `Xnoise13` ou `Xnoise21`. Cette présence de faux positifs reflète une limite classique du LASSO : lorsqu'il opère une sélection agressive, notamment dans un contexte où le signal est fort, il peut retenir certaines variables qui ne contribuent pas réellement au modèle. Cela illustre son instabilité en présence de bruit ou de corrélations fortes.

Elastic Net sélectionne un ensemble très proche de celui du LASSO, mais avec une différence importante : il retient nettement moins de variables purement bruitées. Dans les résultats obtenus, il conserve toutes les variables pertinentes (bloc corrélé et variables fortes), mais seulement une ou deux variables non informatives. Cela montre qu'il est plus robuste que le LASSO face à la sélection de faux positifs. Cet effet stabilisateur provient de la composante Ridge dans la pénalisation mixte, qui aide le modèle à mieux gérer les groupes de variables corrélées et à éviter les fluctuations excessives dans la sélection des coefficients. Ainsi, bien que le LASSO et Elastic Net paraissent sélectionner des ensembles similaires, Elastic Net produit une sélection plus propre et plus cohérente.

Enfin, Ridge constitue le cas extrême opposé. La méthode ne supprime aucune variable : l'ensemble des 30 prédicteurs reçoit un coefficient non nul, y compris toutes les variables purement bruitées. Ridge stabilise les coefficients en présence de corrélations, mais ne réalise aucune sélection. Cette propriété en fait un modèle extrêmement robuste et régulier, mais totalement inadapté lorsqu'il s'agit d'identifier les prédicteurs pertinents. Sur ce jeu simulé, Ridge restitue une vision lissée du signal, mais ne permet pas de distinguer les variables réellement actives des variables sans effet.

Dans l'ensemble, ces résultats montrent que le LASSO est la méthode la plus parcimonieuse mais aussi la plus instable, qu'Elastic Net offre le meilleur compromis entre sélection et robustesse, et que Ridge se distingue par sa stabilité au prix de l'absence totale de sélection. Ce comportement général observé sur données simulées se retrouve également, dans une certaine mesure, dans les analyses menées sur les trois jeux de données réelles du rapport.

Synthèse comparative

Les trois méthodes de pénalisation étudiées diffèrent à la fois dans leur formulation mathématique, dans leur impact sur les données, et dans les résultats obtenus dans les trois cadres considérés : régression linéaire, régression logistique et modèle de Cox.

Formulation mathématique

LASSO, Ridge et Elastic Net partagent l'objectif de régulariser les coefficients afin de limiter le sur-ajustement. Le LASSO repose sur une pénalisation L1 qui pousse certains coefficients à zéro, réalisant ainsi une sélection automatique de variables. Ridge utilise une pénalisation L2 qui réduit les coefficients sans

jamais les annuler, ce qui assure une forte stabilité, notamment en présence de variables corrélées. Elastic Net combine ces deux pénalisations via un paramètre α, permettant à la fois une sélection modérée et une bonne stabilité lorsque des groupes de prédicteurs sont corrélés.

Effets sur les données

Les trois approches diffèrent dans leur manière d'interagir avec les prédicteurs. Le LASSO favorise la parcimonie : il conserve les variables les plus informatives mais peut éliminer certaines variables pourtant pertinentes en cas de corrélations fortes. Ridge adopte une approche opposée : il garde toutes les variables, réduit leurs coefficients et stabilise le modèle même lorsque plusieurs prédicteurs sont colinéaires. Elastic Net combine ces comportements : il peut éliminer certaines variables tout en conservant la structure des groupes corrélés, ce qui en fait une méthode polyvalente dans des contextes où l'information est répartie entre plusieurs prédicteurs.

Résultats obtenus

Les analyses empiriques illustrent ces différences. Dans la régression linéaire, le dataset Boston présente de fortes corrélations : Ridge stabilise efficacement les coefficients, tandis que le LASSO peut supprimer des prédicteurs informatifs. Elastic Net maintient un équilibre entre sélection et stabilité, ce qui lui permet d'obtenir de bonnes performances. En régression logistique, où les classes sont déjà bien séparées, les trois méthodes atteignent des performances similaires ; la pénalisation influe surtout sur la structure du modèle plutôt que sur sa précision. Dans le modèle de Cox, où de nombreuses covariables ont un effet modeste, Ridge et Elastic Net conservent une structure cohérente du risque, tandis que le LASSO peut se montrer trop agressif dans la suppression de variables contributives.

Ainsi, les comportements observés sur les données réelles sont cohérents avec les propriétés théoriques : LASSO privilégie la sélection, Ridge la stabilité, et Elastic Net offre un compromis robuste et équilibré.

Conclusion

L'objectif de ce rapport était de comparer les méthodes de pénalisation LASSO, Ridge et Elastic Net dans trois contextes statistiques distincts : la régression linéaire, la régression logistique et le modèle de Cox. Ces méthodes introduisent un terme de régularisation permettant de contrôler la complexité du modèle, d'améliorer sa stabilité et, pour LASSO et Elastic Net, de sélectionner automatiquement les prédicteurs les plus pertinents.

Les résultats montrent que leurs performances dépendent fortement de la structure des données. Dans le dataset Boston, la présence de corrélations importantes favorise Ridge et Elastic Net, qui stabilisent les estimations sans supprimer des variables utiles. Le LASSO, plus agressif, peut éliminer certains prédicteurs pertinents. Dans la régression logistique, où les classes sont bien séparées, les trois méthodes atteignent des niveaux de performance comparables, et la pénalisation influence surtout la structure du modèle. Dans le modèle de Cox, où les effets sont plus faibles et diffus, Ridge et Elastic Net permettent de mieux capter la structure du risque, tandis que le LASSO peut être trop restrictif.

Globalement, le choix de la méthode dépend du compromis recherché entre parcimonie et stabilité. LASSO est adapté lorsque la réduction du nombre de variables est prioritaire. Ridge est à privilégier lorsque la stabilité des coefficients et la gestion des corrélations sont essentielles. Elastic Net se révèle souvent le choix le plus polyvalent en combinant équilibre, robustesse et capacité de sélection.

Ces résultats confirment que les méthodes de pénalisation ne se limitent pas à améliorer la performance prédictive, mais constituent des outils clés pour structurer et interpréter les modèles statistiques modernes.

Limites et améliorations

Les résultats présentés reposent sur des jeux de données de taille modérée et relativement propres. Dans des contextes plus bruités ou plus dimensionnels, la régularisation pourrait jouer un rôle plus central et révéler davantage de différences entre les méthodes. Une piste d'amélioration consisterait à explorer plusieurs valeurs du paramètre α dans l'Elastic Net afin d'évaluer plus finement le compromis entre pénalisation L1 et L2. Enfin, l'étude pourrait être étendue à des méthodes non linéaires, comme les forêts aléatoires ou le boosting, afin de comparer le comportement des techniques de régularisation dans des cadres plus flexibles.