Rapport de TP - Modèles de Régression Régularisée

Tommy MORALES, Aslan MARTIN

2025-10-19

Table des matières

1 Introduction

Ce rapport, réalisé dans le cadre du cours de Modèles de Régression Régularisée, se concentre sur l'analyse d'un jeu de données concernant la production d'électricité au Mexique. L'objectif est de construire un modèle de régression linéaire permettant d'expliquer et de prédire la consommation énergétique journalière en fonction de variables météorologiques et calendaires. Une attention particulière sera portée à l'identification et au traitement des problèmes de multicolinéarité afin de garantir la robustesse et l'interprétabilité du modèle final.

2 Étude IV : Production d'Électricité au Mexique

2.1 Analyse Exploratoire des Données (EDA)

2.1.1 Problématique et présentation des données

L'objectif est d'expliquer la production d'électricité journalière (*Total*) au Mexique en utilisant un ensemble de variables météorologiques et temporelles. Une rapide analyse confirme qu'aucune donnée n'est manquante, qu'il n'y a pas d'outliers, et que les formats sont consistants. Le csv contient les données de 12 variables collectées pendant 1461 jours.

```
##      X0       RH     SSRD    STRD      T2M   T2Mmax    T2Mmin Covid Holidays
## 1 17897 54.88692 556483.6 1095579 13.61629 20.25525  8.480851     0        1
## 2 17898 61.36585 551500.0 1148553 13.64353 19.71474  9.270146     0        0
## 3 17899 60.81884 643224.1 1086882 13.25186 20.81150  7.662988     0        0
## 4 17900 55.19776 661235.1 1069402 14.09160 22.81910  7.496520     0        0
## 5 17901 54.77641 609757.8 1110888 15.24040 23.64965  8.615337     0        0
## 6 17902 63.39149 548361.9 1207957 16.86465 23.43387 12.072207     0        0
##   DOW TOY    Total
## 1   1   1 588.3452
## 2   2   2 736.1832
```

```
## 3   3   3 793.0117
## 4   4   4 800.6797
## 5   5   5 762.7347
## 6   6   6 690.9642
```

Voici un aperçu des variables présentes dans le jeu de données :

TABLE 1: Description des variables du jeu de données.

Variable	Description
X0	Date (numérique)
RH	Humidité relative (%)
SSRD	Radiation solaire de surface (J.m-2)
STRD	Radiation thermique de surface (J.m-2)
T2M	Température moyenne à 2m (°C)
T2Mmax	Température maximale à 2m (°C)
T2Mmin	Température minimale à 2m (°C)
Covid	Indice de restriction COVID
Holidays	Jours fériés (1) ou non (0)
DOW	Jour de la semaine (0=Lundi, 6=Dimanche)
TOY	Jour de l'année (1-366)
Total	Électricité produite (GWh)

2.1.2 Analyse univariée et bivariée

Une première visualisation de la consommation en fonction de la température minimale montre une forte corrélation positive. Plus la température minimale est élevée, plus la consommation est importante, probablement à cause de la climatisation.

```
plot(df$T2Mmin, df$Total,
     main = "Consommation vs Température Minimale",
     xlab = "Température minimale (°C)",
     ylab = "Consommation totale (GWh)",
     pch = 16, col = "darkblue",
     panel.first = grid())
```

2.1.3 Détection de la Multicolinéarité

Avant la modélisation, une analyse de la corrélation est essentielle pour éviter les problèmes de multicolinéarité qui rendent les modèles instables.

```
corrplot(cor(df))
```

La matrice de corrélation confirme que les variables `T2M`, `T2Mmax`, `T2Mmin` et `STRD` sont très fortement corrélées entre elles (coefficients > 0.85). Les inclure simultanément dans un modèle OLS rendrait l'interprétation des coefficients impossible et non fiable. Elle révèle aussi que les marqueurs temporels et les effets du COVID sont faiblement corrélés avec le Total, et donc qu'ils ne sont peut-être pas essentielles.

Pour vérifier la multicolinéarité, et démontrer les difficultés d'interprétations, observons la relation prédite entre $Total$ et $T2Mmin$ via OLS naïve, avec en noir la courbe pour une relation Total~T2Mmin:

```
model <- lm(Total ~ ., data = df)

plot(df$T2Mmin, df$Total,
     main = "Consommation vs Température Minimale",
     xlab = "Température minimale (°C)",
```

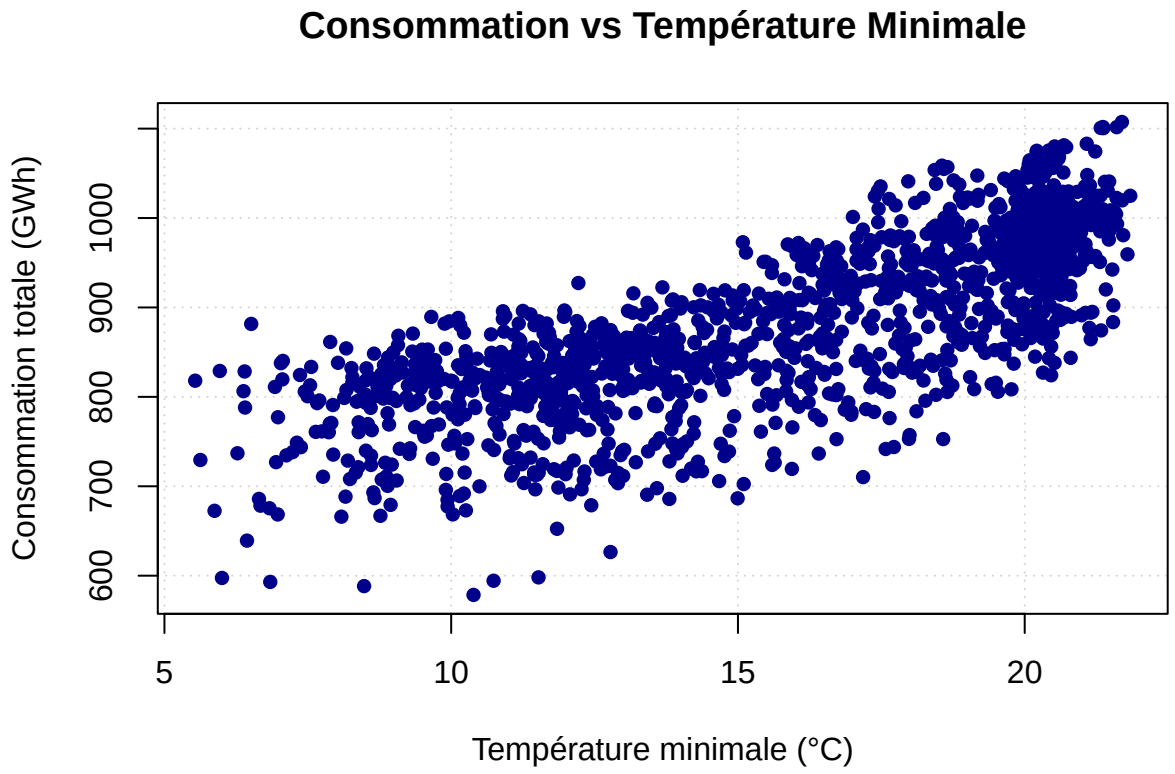


FIGURE 1 – Consommation en fonction de la température minimale.

```
     ylab = "Consommation totale (GWh)",
     pch = 16, col = "darkblue",
     panel.first = grid())

T2Mmin_range <- seq(min(df$T2Mmin), max(df$T2Mmin), length.out = 100)

predict_data_2 <- data.frame(
  T2Mmin = T2Mmin_range,
  X0 = mean(df$X0),
  RH = mean(df$RH),
  SSRD = mean(df$SSRD),
  STRD = mean(df$STRD),
  T2M = mean(df$T2M),
  Covid = mean(df$Covid),
  T2Mmax = mean(df$T2Mmax),
  Holidays = mean(df$Holidays),
  DOW = mean(df$DOW),
  TOY = mean(df$TOY)
)

total_predictions_2 <- predict(model, predict_data_2)

lines(T2Mmin_range, total_predictions_2, col = "red", lwd = 2)

model_check <- lm(Total ~ T2Mmin, data = df)
```

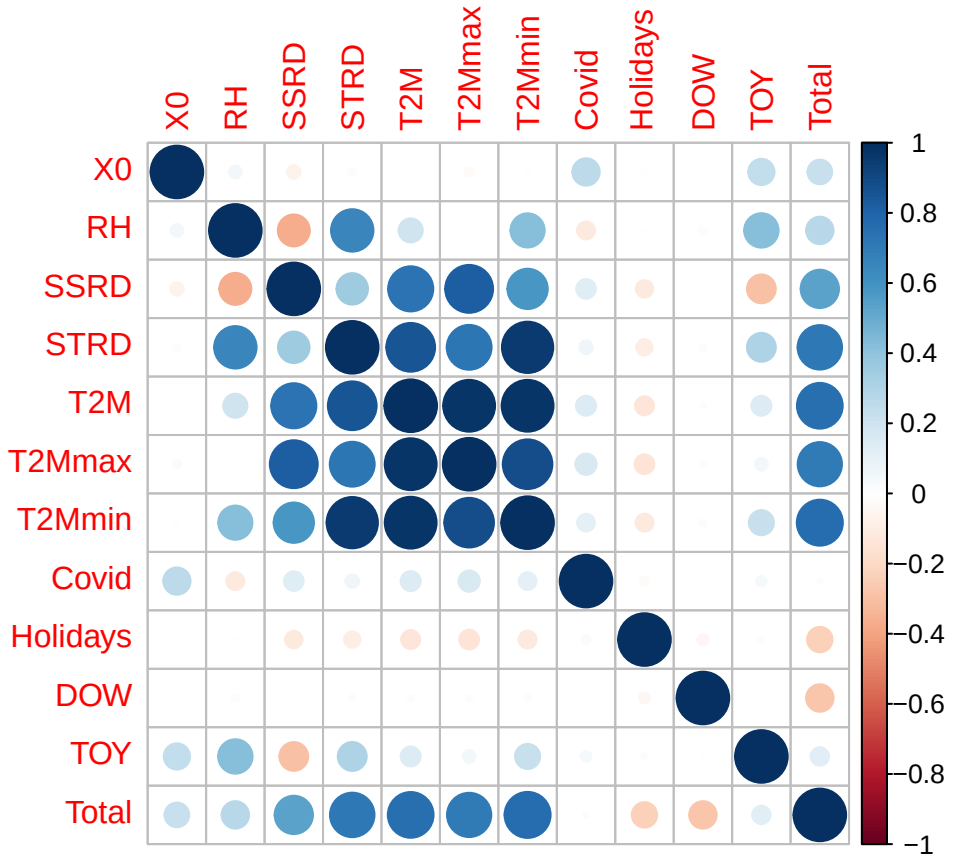


FIGURE 2 – Matrice de corrélation des variables.

```r
abline(model_check, col = "black", lwd = 2)

legend("topleft", legend = "Prédiction naïve", col = "red", lwd = 2)
```

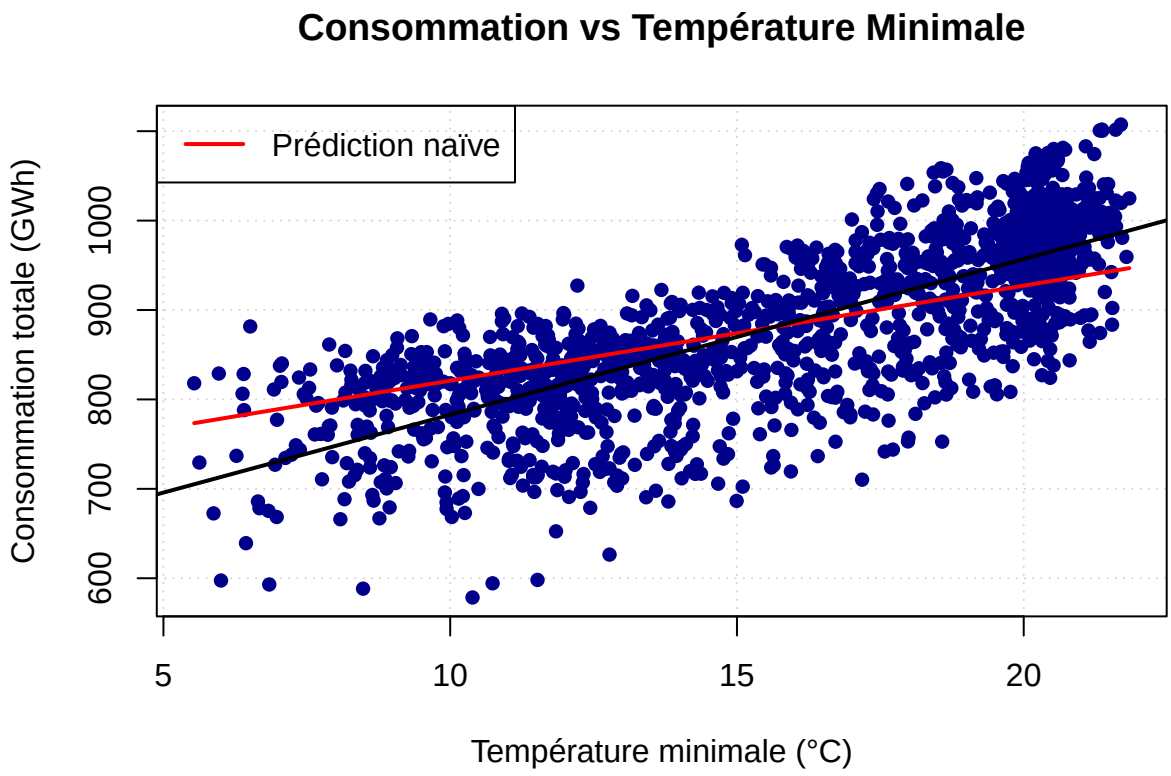


FIGURE 3 – Consommation en fonction de la température minimale prédite naïvement.

Les prédictions sont imparfaites (visuellement parlant) du fait de l'influence des autres variables corrélées (telles que T2Mmax). Dès lors, l'interprétation des résultats devient difficile.

2.2 Modélisation et Sélection de Variables

2.2.1 Premier modèle : OLS avec toutes les variables

Nous construisons un premier modèle incluant toutes les variables pour observer les effets de la multicolinéarité.

```r
model1 <- lm(Total ~ ., data = df)
summary(model1)
```

```
##
## Call:
## lm(formula = Total ~ ., data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -154.97  -30.11    5.53   32.72  169.06
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -8.928e+02  9.916e+01  -9.003  < 2e-16 ***
## X0           6.378e-02  3.067e-03  20.797  < 2e-16 ***
```

```
## RH          -1.048e+00  4.129e-01  -2.538   0.0113 *
## SSRD         1.443e-04  1.981e-05   7.285 5.26e-13 ***
## STRD         4.477e-04  8.249e-05   5.427 6.73e-08 ***
## T2M         -9.645e+00  7.474e+00  -1.291   0.1971
## T2Mmax       2.368e+00  3.833e+00   0.618   0.5368
## T2Mmin       1.063e+01  5.133e+00   2.071   0.0385 *
## Covid       -5.478e-01  4.535e-02 -12.078  < 2e-16 ***
## Holidays    -8.783e+01  7.073e+00 -12.416  < 2e-16 ***
## DOW         -1.297e+01  6.001e-01 -21.615  < 2e-16 ***
## TOY         -2.359e-02  1.548e-02  -1.524   0.1278
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 45.61 on 1449 degrees of freedom
## Multiple R-squared:  0.7716, Adjusted R-squared:  0.7699
## F-statistic: 445.1 on 11 and 1449 DF,  p-value: < 2.2e-16
```

Ce modèle présente les problèmes attendus : des variables de température comme $T2M$ et $T2Mmax$ sont inexploitables, et le coefficient de $T2M$ est négatif, ce qui est contre-intuitif (Si T2M et T2Mmin ont des coefficients positifs, la logique veut que ce soit aussi le cas de T2M). Le modèle est donc instable et peu interprétable (voir le plot précédent).

2.2.2 Interprétation du modèle final

Pour construire un modèle robuste, nous sélectionnons un sous-ensemble de variables en éliminant celles qui sont redondantes. Nous conservons `T2Mmin` comme unique représentant de la température et retirons `T2M`, `T2Mmax` et `STRD`. Nous enlevons également `TOY`, qui est équivalent à `X0`.

```
model_final <- lm(Total ~ X0 + RH + SSRD + T2Mmin + Covid + Holidays + DOW, data = df)
summary(model_final)
```

```
##
## Call:
## lm(formula = Total ~ X0 + RH + SSRD + T2Mmin + Covid + Holidays +
##     DOW, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -142.635  -29.676    5.065   32.993  185.851
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.582e+02  5.881e+01  -9.491  < 2e-16 ***
## X0           6.099e-02  3.005e-03  20.295  < 2e-16 ***
## RH           7.344e-01  2.640e-01   2.781  0.00548 **
## SSRD         1.284e-04  1.574e-05   8.159 7.22e-16 ***
## T2Mmin       1.330e+01  6.964e-01  19.102  < 2e-16 ***
## Covid       -5.296e-01  4.593e-02 -11.531  < 2e-16 ***
## Holidays    -8.664e+01  7.151e+00 -12.116  < 2e-16 ***
## DOW         -1.290e+01  6.068e-01 -21.257  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 46.28 on 1453 degrees of freedom
## Multiple R-squared:  0.7642, Adjusted R-squared:  0.7631
```

```
## F-statistic: 672.9 on 7 and 1453 DF,  p-value: < 2.2e-16
```

Ce modèle final est bien plus satisfaisant : toutes les variables sont très significatives et les signes des coefficients sont logiques. Le R-squared ajusté de **0.7631** signifie que notre modèle explique environ **76.3%** de la variance de la consommation totale.

L'interprétation des coefficients nous apprend que:

- *T2Mmin* : Une augmentation de 1°C de la température minimale est associée à une hausse de **13.30 GWh** de la consommation.
- *Holidays* : Un jour férié réduit la consommation de **86.64 GWh** par rapport à un jour ouvré.
- *DOW* : Chaque jour qui passe dans la semaine (de Lundi=0 à Dimanche=6) est associé à une baisse moyenne de **12.90 GWh**.
- *Covid* : Un point d'indice de restriction COVID en plus est associé à une baisse de **0.5 GWh** de la consommation, ce qui est cohérent avec une baisse de l'activité économique.

```r
plot(df$T2Mmin, df$Total,
     main = "Consommation vs Température Minimale",
     xlab = "Température minimale (°C)",
     ylab = "Consommation totale (GWh)",
     pch = 16, col = "darkblue",
     panel.first = grid())

T2Mmin_range <- seq(min(df$T2Mmin), max(df$T2Mmin), length.out = 100)

predict_data_3 <- data.frame(
  T2Mmin = T2Mmin_range,
  X0 = mean(df$X0),
  RH = mean(df$RH),
  SSRD = mean(df$SSRD),
  Covid = mean(df$Covid),
  Holidays = mean(df$Holidays),
  DOW = mean(df$DOW)
)

total_predictions_3 <- predict(model_final, predict_data_3)

lines(T2Mmin_range, total_predictions_3, col = "red", lwd = 2)

model_check <- lm(Total ~ T2Mmin, data = df)

abline(model_check, col = "black", lwd = 2)

legend("topleft", legend = "Prédiction manuellement filtrée", col = "red", lwd = 2)
```

On observe que l'angle s'est approché de la projection théorique, ce qui était attendu: La multicolinéarité n'impose plus de compromis entre les variables T2M et STRD, forçant T2Mmin à être plus représentative, et donc interprétable.

2.3 Détermination d'un modèle plus fiable par Best Subset

Notre modèle précédent visait à nous débarasser des variables parasites évidentes. Le choix rationnel pour poursuivre notre étude est donc de généraliser cette méthode, et de vérifier que toutes les variables encore présentes sont aussi utiles qu'il y parrait.

Pour cela, nous utilisons une méthode de Best Subset, implémentée en itérant sur toutes les combinaisons (plutôt qu'un algorithme type branch and bound) par soucis de simplicité, et car nous n'avons que 11 variables

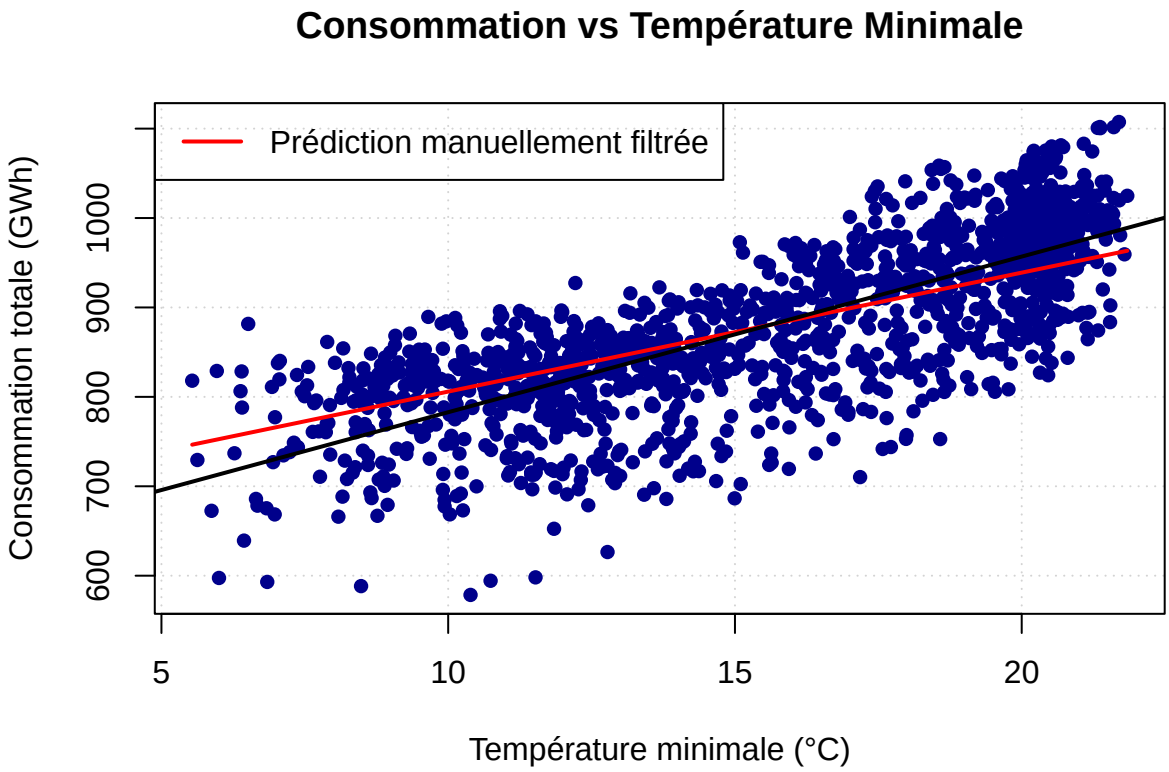


FIGURE 4 – Consommation en fonction de la température minimale prédite après filtrage des variables

(ce qui est suffisament faible pour rendre une implémentation Brute Force tout à fait viable).

```
predictors <- setdiff(names(df), "Total")
n_predictors <- length(predictors)
results_list <- list()
model_counter <- 1 # Un compteur pour suivre les modèles
for (k in 1:n_predictors) {

  combinations <- combn(predictors, k, simplify = FALSE)

  for (combo in combinations) {

    formula_rhs <- paste(combo, collapse = " + ") # Pour avoir les variables de lm
    formula <- as.formula(paste("Total ~", formula_rhs))

    model <- lm(formula, data = df)

    model_summary <- summary(model) # Pour récupérer le R² ajusté
    aic_score <- AIC(model)
    bic_score <- BIC(model)
    adjr2_score <- model_summary$adj.r.squared

    results_list[[model_counter]] <- data.frame(
      num_variables = k,
      variables = paste(combo, collapse = ", "),
      AIC = aic_score,
      BIC = bic_score,
```

```
      adjR2 = adjr2_score
    )

    model_counter <- model_counter + 1
  }
}
```

```
## [1] "Meilleur subset via AIC: "
```

```
## [1] "X0, RH, SSRD, STRD, T2M, T2Mmin, Covid, Holidays, DOW, TOY"
```

```
## AIC: 15320.84
```

```
## [1] "Meilleur subset via R² ajusté: "
```

```
## [1] "X0, RH, SSRD, STRD, T2M, T2Mmin, Covid, Holidays, DOW, TOY"
```

```
## R²: 0.7700073
```

Sans surprise, l'absence de pénalités du R^2 ajusté et les pénalités trop faibles du calcul de AIC amènent à trouver un subset contenant l'ensemble des variables.

Pour un résultat plus fiables, il faut regarder le résultat obtenu avec le calcul de BIC

```
## [1] "Meilleur subset via BIC: "
```

```
## [1] "X0, SSRD, STRD, Covid, Holidays, DOW"
```

```
## BIC: 15368.52
```

Comme attendu, plusieurs variables ont disparu:

- **TOY** qui est redondant avec X0
- **Les variables corrélée du groupe SSRD-T2Mmin**, même s'il est surprenant de voir que le meilleur subset a priviligié STRD plutôt que T2Mmin
- **RH** dont le summary du modèle précédent révélait déjà les faiblesses (en terme de potentiel d'interprétation de $Total$)

Ce modèle est donc parfaitement cohérent, et offre une meilleure robustesse que le précédent, qui dépendait d'une annalyse arbitraire.

Cependant, la méthode d'utiliser un subset de variable crée une perte d'information. Nous allons donc essayer des méthodes plus généralistes, qui permettent de limiter l'influence des variables redondantes.

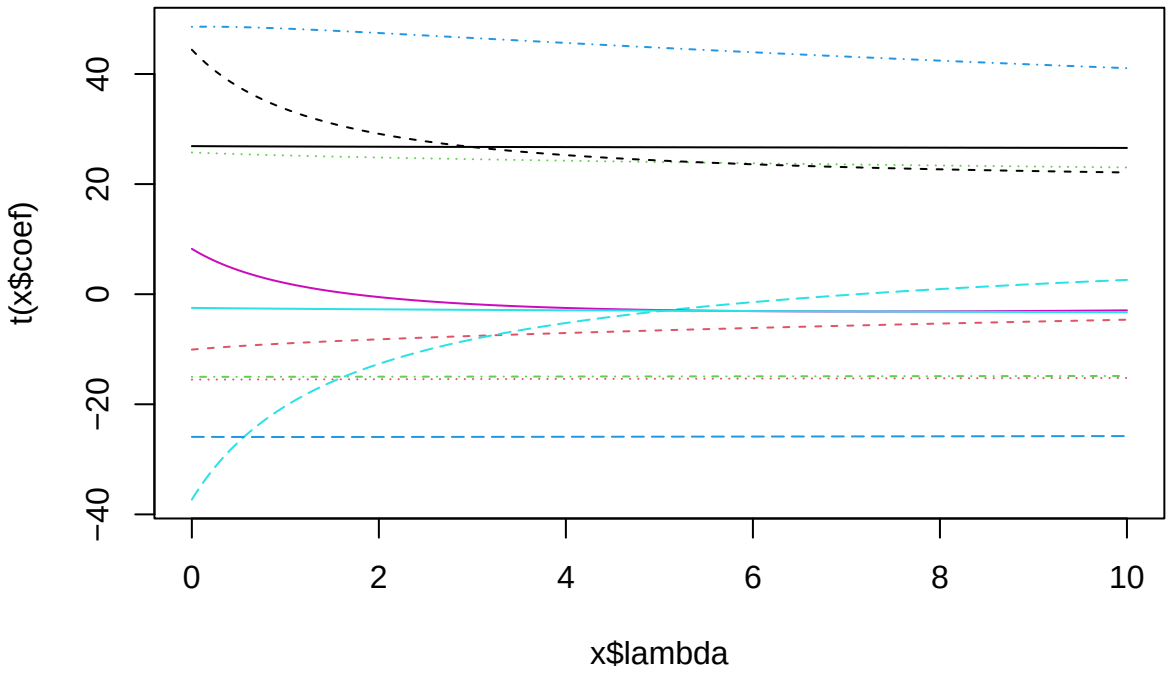
2.4 La Régression Ridge pour une généralisation

Pour cela, il existe la Régression Ridge.

$\hat{\beta} = argmin_\beta ||Y - X\beta||_2^2 + \lambda||\beta||_2^2$

Au lieu de nous débarasser de certaines variables, cette méthode les préserve toutes, mais va pénaliser les coefficients trop larges. Dès lors, les coefficients de nos variables corrélées ne seront plus gérés de manière imprévisible, mais le poids de chacune sera réduit, ce qui rendra le modèle plus stable.

```
model_ridge <- lm.ridge(Total~., data=df, lambda = seq(0, 10, by = 0.01))
plot(model_ridge)
```




```
MASS::select(model_ridge)
```

```
## modified HKB estimator is 2.222448
## modified L-W estimator is 2.685486
## smallest value of GCV  at 2.19
```

Le graphique des traces est parlant : plus on pénalise (lambda augmente), plus les coefficients se rapprochent vers zéro. Le modèle est donc stabilisé. D'après select, le meilleur *lambda* est **2.19**. Avec cette valeur, on a un modèle régularisé, mais qui garde toutes ses variables, ce qui le rend moins simple à interpréter qu'un subset, mais le rend plus robuste et fiable.

2.5 La Régression Lasso pour faire le tri automatiquement

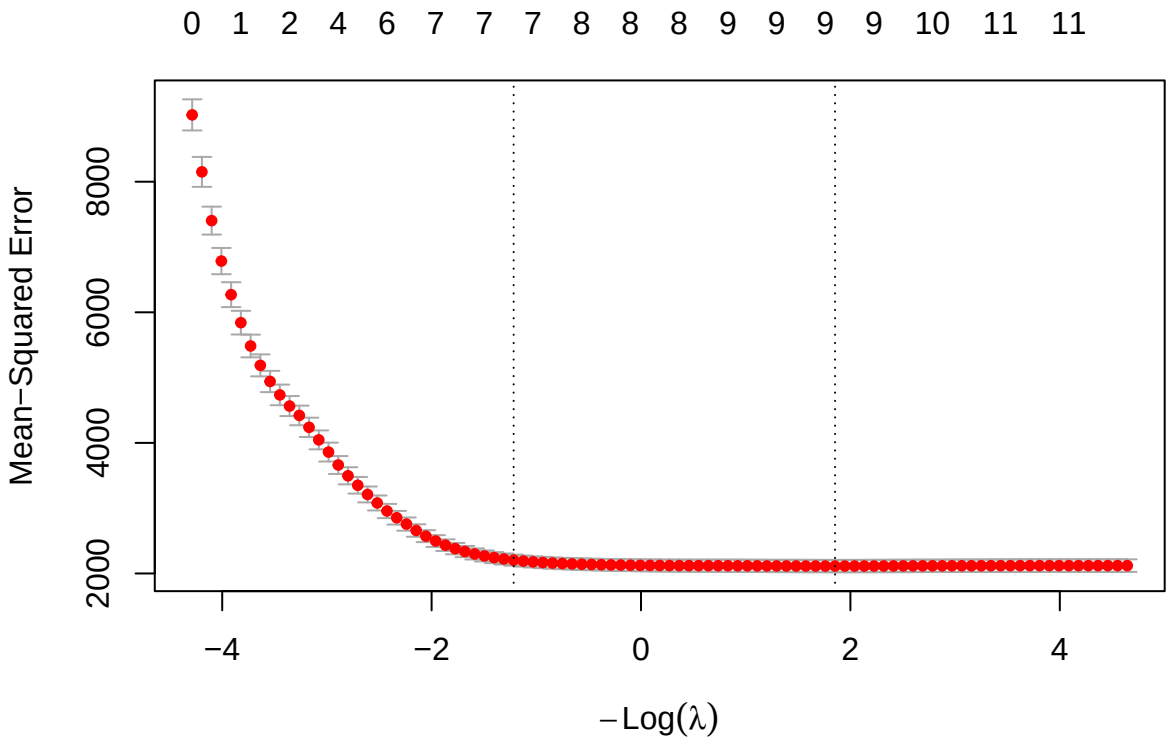
Le problème avec la méthode Ridge est que l'on se retrouve avec toutes les variables, même celles qui pourraient s'avérer inutiles (comme TOY). Nous allons employer la régression Lasso pour cela, qui a pour avantage d'introduire une pénalité capable d'attendre des coefficients nuls, et pas simplement d'y converger.

$\hat{\beta} = argmin_{\beta}||Y - X\beta||_2^2 + \lambda||\beta||_1$

```
x <- model.matrix(Total ~ . -1, data = df)
y <- df$Total

cv_model_lasso <- glmnet::cv.glmnet(x, y, alpha = 1)

# Afficher le graphique
plot(cv_model_lasso)
```




```
# Extraire les lambdas
lambda_lasso <- cv_model_lasso$lambda.1se

# Afficher les coefficients
final_coeffs_lasso <- coef(cv_model_lasso, s = lambda_lasso)
print(final_coeffs_lasso)
```

```
## 12 x 1 sparse Matrix of class "dgCMatrix"
##                 s=3.375891
## (Intercept) -5.079335e+02
## X0           4.963151e-02
## RH           .
## SSRD         1.167115e-04
## STRD         2.320649e-04
## T2M          .
## T2Mmax       .
## T2Mmin       7.741806e+00
## Covid       -3.472558e-01
## Holidays    -6.830762e+01
## DOW         -1.119981e+01
## TOY          .
```

Il semble donc que le modèle proposé par la régréssion Lasso utilise les variables {X0, SSRD, STRD, T2Mmin, COVID, Holidays, DOW}

Comme attendu, on obtient un résultat très proche de celui obtenu par la méthode du Best Subset (Seule T2Mmin diffère). Cependant, quand la méthode du Best Subset est peu robuste, cette méthode s'adaptera de manière très fluide à l'expension de notre Jeu de donnée.

3 Comparaison des Modèles et Conclusion Finale

On a donc testé quatre manières de faire :

Approche	Sélection Variables	Gestion Multilinéarité	Idéal Pour...
OLS Manuel	Manuelle	Suppression	Interpréter les coeffs
Best Subset (BIC)	Systématique	Suppression	Trouver le modèle le plus simple
Ridge	Aucune (garde tout)	Réduit l'influence	Prédiction stable
Lasso	Automatique (met à zéro)	Réduit et supprime	Mix prédiction et simplicité

Pour conclure, notre analyse nous a permis de construire un modèle solide. Si le but est de **comprendre** quels facteurs jouent un rôle, notre **modèle OLS final** ou celui du **Best Subset (BIC)** sont parfaits. Ils sont simples, logiques, et expliquent bien les données.

Si le but est d'avoir le **meilleur prédicteur** possible, le **Lasso** est probablement le gagnant. Il fait un tri intelligent tout seul et régularise le reste, ce qui le rend plus robuste face à de nouvelles données. Il confirme de plus les résultats obtenus par la méthode du Best Subset.

De cette manière, les méthodes **OLS filtrée** et **Best Subset** sont de bons outils pour guider notre collecte de donnée, et leur filtrage, tandis que la méthode **Lasso** sera le meilleur de nos outils pour exploiter ces données afin d'émettre des prédictions.

Pour démontrer ce résultat, il aurait fallu diviser notre data_set pour avoir des données d'entraînement et de test, et vérifier la robustesse de nos modèles en:

— Entraînant chaque modèle sur notre Training Set
— Déterminant le taux d'erreur de chaque modèle sur notre Test Set

On aurait alors remarqué que les méthodes Lasso ou Ridge offrent les meilleurs résultats.

Cependant, cela dépasse le cadre de ce TP et de notre maîtrise du language R, donc nous nous arrêterons là.