TP Régression linéaire

Simon Desdevises

03/10/2025

```
data <- read.csv(
  "/Users/simondesdevises/Documents/Polytech/IS2A/S7/RL/TP3/data.csv",
  header = TRUE
)
rownames(data) <- data$Y
```

1.1 Statistiques descriptives univariées

Statistiques descriptives de toutes les variables

```
# Statistiques descriptives complètes
knitr::kable(summary(data[, 1:7]))
```

X1	X2	X3	X4	X5	X6	X7
Min. :0.00000	Min. :0.0000	Min. :0.00000	Min. :0.0000	Min. :0.00000	Min. :0.0000	Min. :0.01000
1st Qu.:0.00000	1st Qu.:0.0750	1st Qu.:0.00000	1st Qu.:0.0000	1st Qu.:0.00000	1st Qu.:0.0600	1st Qu.:0.03750
Median :0.00000	Median :0.2000	Median :0.00000	Median :0.3150	Median :0.01000	Median :0.2750	Median :0.07000
Mean :0.07417	Mean :0.2183	Mean :0.04333	Mean :0.2533	Mean :0.04333	Mean :0.3108	Mean :0.05667
3rd Qu.:0.17000	3rd Qu.:0.2925	3rd Qu.:0.10000	3rd Qu.:0.3800	3rd Qu.:0.12000	3rd Qu.:0.4625	3rd Qu.:0.08000
Max. :0.21000	Max. :0.6200	Max. :0.12000	Max. :0.6200	Max. :0.12000	Max. :0.7400	Max. :0.08000

1.2 Statistiques descriptives bivariées

Matrice de corrélation

```
# Matrice de corrélation
cor_matrix <- cor(data)
print("Matrice de corrélation :")
```

```
## [1] "Matrice de corrélation :"
```

```r
print(round(cor_matrix, 3))
```

```
##        X1     X2     X3     X4     X5     X6     X7      Y
## X1  1.000  0.104  1.000  0.371 -0.548 -0.805  0.603 -0.837
## X2  0.104  1.000  0.101 -0.537 -0.293 -0.191 -0.590 -0.071
## X3  1.000  0.101  1.000  0.374 -0.548 -0.805  0.607 -0.838
## X4  0.371 -0.537  0.374  1.000 -0.211 -0.646  0.916 -0.707
## X5 -0.548 -0.293 -0.548 -0.211  1.000  0.463 -0.274  0.494
## X6 -0.805 -0.191 -0.805 -0.646  0.463  1.000 -0.656  0.985
## X7  0.603 -0.590  0.607  0.916 -0.274 -0.656  1.000 -0.741
## Y  -0.837 -0.071 -0.838 -0.707  0.494  0.985 -0.741  1.000
```

```r
# Corrélations de Y avec les autres variables
print("Corrélations de Y avec les variables explicatives :")
```

```
## [1] "Corrélations de Y avec les variables explicatives :"
```

```r
cor_y <- cor(data)[, "Y"]
print(sort(cor_y, decreasing = TRUE))
```

```
##           Y          X6          X5          X2          X4          X7
##  1.00000000  0.98507041  0.49379905 -0.07081888 -0.70671354 -0.74111624
##          X1          X3
## -0.83729576 -0.83795781
```

Analyse détaillée des relations bivariées

```r
cat("\n=== ANALYSE BIVARIÉE DÉTAILLÉE ===\n\n")
```

```
##
## === ANALYSE BIVARIÉE DÉTAILLÉE ===
```

```r
for (i in seq_len(ncol(data[, 1:7]))) {
  var_name <- colnames(data)[i]
  cat("\n--- Variable:", var_name, "---\n")

  cor_val <- cor(data[, i], data$Y)
  cat("Corrélation avec Y:", round(cor_val, 4), "\n")

  model <- lm(Y ~ data[, i], data = data)
  cat("R² de la régression simple:", round(summary(model)$r.squared, 4), "\n")
  cat("p-value:", format.pval(summary(model)$coefficients[2, 4]), "\n")
}
```

```
##
## --- Variable: X1 ---
## Corrélation avec Y: -0.8373
## R² de la régression simple: 0.7011
```

```
## p-value: 0.00067872
##
## --- Variable: X2 ---
## Corrélation avec Y: -0.0708
## R² de la régression simple: 0.005
## p-value: 0.82688
##
## --- Variable: X3 ---
## Corrélation avec Y: -0.838
## R² de la régression simple: 0.7022
## p-value: 0.00066581
##
## --- Variable: X4 ---
## Corrélation avec Y: -0.7067
## R² de la régression simple: 0.4994
## p-value: 0.01018
##
## --- Variable: X5 ---
## Corrélation avec Y: 0.4938
## R² de la régression simple: 0.2438
## p-value: 0.10276
##
## --- Variable: X6 ---
## Corrélation avec Y: 0.9851
## R² de la régression simple: 0.9704
## p-value: 5.6971e-09
##
## --- Variable: X7 ---
## Corrélation avec Y: -0.7411
## R² de la régression simple: 0.5493
## p-value: 0.0058156
```

2. Régression linéaire multiple

Modèle complet avec toutes les variables

```r
# Régression linéaire multiple : Y en fonction de toutes les variables X1 à X7
model_complet <- lm(Y ~ X1 + X2 + X3 + X4 + X5 + X6 + X7, data = data)

# Affichage du résumé du modèle
cat("\n=== MODÈLE DE RÉGRESSION LINÉAIRE MULTIPLE ===\n\n")
```

```
##
## === MODÈLE DE RÉGRESSION LINÉAIRE MULTIPLE ===
```

```r
print(summary(model_complet))
```

```
##
## Call:
## lm(formula = Y ~ X1 + X2 + X3 + X4 + X5 + X6 + X7, data = data)
##
```

```
## Residuals:
##       98.7       97.8       96.6         92       86.6       91.2       81.9
##  1.207e+00 -2.218e-01 -5.475e-01 -8.195e-02  8.105e-01 -3.527e-01  5.906e-02
##       83.1       82.4       83.2       81.4       88.1
##  3.598e-01 -3.036e-01 -1.153e-01  1.850e-15 -8.141e-01
##
## Coefficients: (1 not defined because of singularities)
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   116.92      81.10   1.442    0.209
## X1            -82.60     173.22  -0.477    0.654
## X2            -31.00      80.84  -0.383    0.717
## X3             24.33     431.97   0.056    0.957
## X4            -39.74      90.25  -0.440    0.678
## X5            -29.17      84.02  -0.347    0.743
## X6            -16.62      84.45  -0.197    0.852
## X7                NA         NA      NA       NA
##
## Residual standard error: 0.8362 on 5 degrees of freedom
## Multiple R-squared:  0.9925, Adjusted R-squared:  0.9836
## F-statistic: 110.7 on 6 and 5 DF,  p-value: 3.762e-05
```

```
# Coefficients du modèle
cat("\n=== COEFFICIENTS DU MODÈLE ===\n")
```

```
##
## === COEFFICIENTS DU MODÈLE ===
```

```
print(round(coef(model_complet), 4))
```

```
## (Intercept)          X1          X2          X3          X4          X5
##    116.9213    -82.6011    -30.9984     24.3307    -39.7412    -29.1711
##          X6          X7
##    -16.6204          NA
```

```
# Qualité de l'ajustement
cat("\n=== QUALITÉ DE L'AJUSTEMENT ===\n")
```

```
##
## === QUALITÉ DE L'AJUSTEMENT ===
```

```
cat("R² :", round(summary(model_complet)$r.squared, 4), "\n")
```

```
## R² : 0.9925
```

```
cat("R² ajusté :", round(summary(model_complet)$adj.r.squared, 4), "\n")
```

```
## R² ajusté : 0.9836
```

```r
cat("Erreur résiduelle :", round(summary(model_complet)$sigma, 4), "\n")
```

```
## Erreur résiduelle : 0.8362
```

```r
cat("Nombre d'observations :", nobs(model_complet), "\n")
```

```
## Nombre d'observations : 12
```

```r
cat("Degrés de liberté :", summary(model_complet)$df[2], "\n")
```

```
## Degrés de liberté : 5
```

On observe que X7 ne peut pas être estimé. Cela est logique car il est corrélé avec les autres variables `X7 = 1 - (X1 + X2 + X3 + X4 + X5 + X6)`.

Constatations

```r
cat("\n=== ANALYSE DES RÉSULTATS ===\n\n")
```

```
##
## === ANALYSE DES RÉSULTATS ===
```

```r
# 1. Analyse de la significativité globale du modèle
f_stat <- summary(model_complet)$fstatistic
p_value_global <- pf(f_stat[1], f_stat[2], f_stat[3], lower.tail = FALSE)
cat("1. SIGNIFICATIVITÉ GLOBALE DU MODÈLE\n")
```

```
## 1. SIGNIFICATIVITÉ GLOBALE DU MODÈLE
```

```r
cat("   F-statistic:", round(f_stat[1], 4), "\n")
```

```
##    F-statistic: 110.6719
```

```r
cat("   p-value:", format.pval(p_value_global), "\n")
```

```
##    p-value: 3.7619e-05
```

```r
if (p_value_global < 0.05) {
  cat("Le modèle est globalement significatif (p < 0.05)\n")
} else {
  cat("Le modèle n'est pas significatif (p >= 0.05)\n")
}
```

```
## Le modèle est globalement significatif (p < 0.05)
```

```r
# 2. Analyse des coefficients individuels
cat("\n2. SIGNIFICATIVITÉ DES VARIABLES INDIVIDUELLES\n")
```

```
##
## 2. SIGNIFICATIVITÉ DES VARIABLES INDIVIDUELLES
```

```r
coef_summary <- summary(model_complet)$coefficients
for (i in 2:nrow(coef_summary)) {
  var_name <- rownames(coef_summary)[i]
  p_val <- coef_summary[i, 4]
  cat("   ", var_name, ":")
  if (p_val < 0.05) {
    cat(" SIGNIFICATIF (p =", format.pval(p_val), ")\n")
  } else {
    cat(" NON significatif (p =", format.pval(p_val), ")\n")
  }
}
```

```
##     X1 : NON significatif (p = 0.65357 )
##     X2 : NON significatif (p = 0.71714 )
##     X3 : NON significatif (p = 0.95726 )
##     X4 : NON significatif (p = 0.67806 )
##     X5 : NON significatif (p = 0.74257 )
##     X6 : NON significatif (p = 0.85173 )
```

Le modèle en lui même est significatif cependant les Variables individuelles ne sont pas significatives. Cela est logique car les variables sont corrélées entre elles.

3. Vérification de la multi-colinéarité

Vérification de la somme des composantes

```r
# Les variables X's représentent les taux de chaque composante dans l'essence
# La somme sur chaque ligne doit faire 100% (ou 1)
cat("=== VÉRIFICATION DE LA SOMME DES COMPOSANTES ===\n\n")
```

```
## === VÉRIFICATION DE LA SOMME DES COMPOSANTES ===
```

```r
# Calculer la somme de X1 à X7 pour chaque ligne
sommes <- apply(data[, 1:7], 1, sum)
cat("Somme X1 + X2 + ... + X7 pour chaque observation :\n")
```

```
## Somme X1 + X2 + ... + X7 pour chaque observation :
```

```r
print(round(sommes, 4))
```

```
## 98.7 97.8 96.6   92 86.6 91.2 81.9 83.1 82.4 83.2 81.4 88.1
##    1    1    1    1    1    1    1    1    1    1    1    1
```

```r
cat("\nToutes les sommes sont égales à 1 :", all(round(sommes, 10) == 1), "\n")
```

```
##
## Toutes les sommes sont égales à 1 : TRUE
```

Calcul du déterminant de X^T X

```r
cat("\n=== DÉTERMINANT DE LA MATRICE X^T X ===\n\n")
```

```
##
## === DÉTERMINANT DE LA MATRICE X^T X ===
```

```r
X <- as.matrix(data[, 1:7])
XTX <- t(X) %*% X

cat("Déterminant de X^T X :", det(XTX), "\n")
```

```
## Déterminant de X^T X : 2.510764e-12
```

```r
if (abs(det(XTX)) < 1e-10) {
  cat("\n→ Le déterminant est nul \n")
  cat("Cela confirme la multi-colinéarité\n")
}
```

```
##
## → Le déterminant est nul
## Cela confirme la multi-colinéarité
```

```r
cat("\n=== CONCLUSION ===\n")
```

```
##
## === CONCLUSION ===
```

```r
cat("On n'a pas besoin des 7 variables puisque 6 suffisent !\n")
```

```
## On n'a pas besoin des 7 variables puisque 6 suffisent !
```

```r
cat("La 7ème variable est automatiquement déterminée par les 6 autres.\n")
```

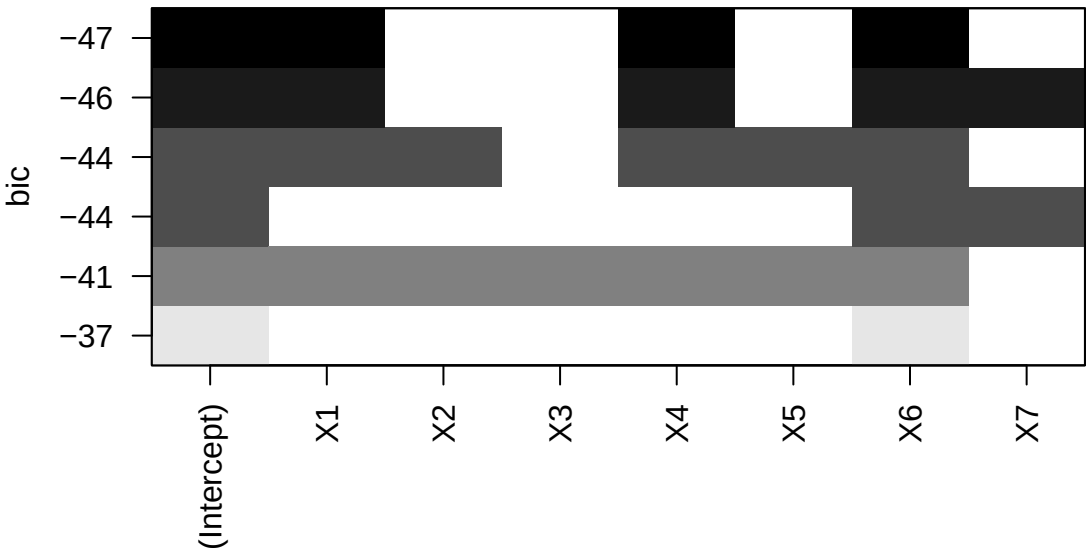
```
## La 7ème variable est automatiquement déterminée par les 6 autres.
```

4 Choix des variables

```r
library(leaps)
choix <- regsubsets(
  Y ~ .,
  int = TRUE,
  nbest = 1,
  nvmax = 7,
  method = "exh",
  data = data
)
res <- summary(choix)
print(res)
```

```
## Subset selection object
## Call: regsubsets.formula(Y ~ ., int = TRUE, nbest = 1, nvmax = 7, method = "exh",
##     data = data)
## 7 Variables  (and intercept)
##    Forced in Forced out
## X1     FALSE      FALSE
## X2     FALSE      FALSE
## X3     FALSE      FALSE
## X4     FALSE      FALSE
## X5     FALSE      FALSE
## X6     FALSE      FALSE
## X7     FALSE      FALSE
## 1 subsets of each size up to 6
## Selection Algorithm: exhaustive
##          X1  X2  X3  X4  X5  X6  X7
## 1  ( 1 ) " " " " " " " " " " "*" " "
## 2  ( 1 ) " " " " " " " " " " "*" "*"
## 3  ( 1 ) "*" " " " " "*" " " "*" " "
## 4  ( 1 ) "*" " " " " "*" " " "*" "*"
## 5  ( 1 ) "*" "*" " " "*" "*" "*" " "
## 6  ( 1 ) "*" "*" "*" "*" "*" "*" " "
```

```r
#choix du meilleur modèle selon le critère BIC
plot(choix, scale = "bic")
```



4-A Identification du meilleur modèle

```
# Identifier le modèle avec le BIC minimum
cat("=== IDENTIFICATION DU MEILLEUR MODÈLE ===\n\n")
```

```
## === IDENTIFICATION DU MEILLEUR MODÈLE ===
```

```
# Trouver le nombre de variables optimal selon BIC
bic_values <- res$bic
meilleur_idx <- which.min(bic_values)
cat("Nombre de variables du meilleur modèle :", meilleur_idx, "\n")
```

```
## Nombre de variables du meilleur modèle : 3
```

```
cat("Valeur du BIC minimum :", round(bic_values[meilleur_idx], 4), "\n\n")
```

```
## Valeur du BIC minimum : -47.2047
```

```
# Afficher les variables sélectionnées
cat("Variables incluses dans le meilleur modèle :\n")
```

```
## Variables incluses dans le meilleur modèle :
```

```r
variables_selectionnees <- names(which(res$which[meilleur_idx, ]))
variables_selectionnees <- variables_selectionnees[variables_selectionnees != "(Intercept)"]
print(variables_selectionnees)
```

```
## [1] "X1" "X4" "X6"
```

Réponse : Le meilleur modèle selon le critère BIC contient **3 variables** : X1, X4, X6.

```r
m2 <- lm(Y ~ X1 + X4 + X6, data = data)
summary(m2)
```

Estimation du meilleur modèle

```
##
## Call:
## lm(formula = Y ~ X1 + X4 + X6, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.00154 -0.41198  0.02205  0.29286  1.00148
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  85.9435     0.9964  86.255 3.64e-13 ***
## X1          -14.0924     4.1175  -3.423  0.00905 **
## X4           -4.9445     1.3018  -3.798  0.00525 **
## X6           15.8852     1.5779  10.067 8.07e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.707 on 8 degrees of freedom
## Multiple R-squared:  0.9915, Adjusted R-squared:  0.9882
## F-statistic: 309.3 on 3 and 8 DF,  p-value: 1.31e-08
```

```r
cat("\n=== ANALYSE DES RÉSULTATS DU MODÈLE 2 ===\n\n")
```

```
##
## === ANALYSE DES RÉSULTATS DU MODÈLE 2 ===
```

```r
f_stat <- summary(m2)$fstatistic
p_value_global <- pf(f_stat[1], f_stat[2], f_stat[3], lower.tail = FALSE)
cat("1. SIGNIFICATIVITÉ GLOBALE DU MODÈLE\n")
```

```
## 1. SIGNIFICATIVITÉ GLOBALE DU MODÈLE
```

```r
cat("   F-statistic:", round(f_stat[1], 4), "\n")
```

```
##    F-statistic: 309.2877
```

```r
cat("   p-value:", format.pval(p_value_global), "\n")
```

```
##    p-value: 1.3095e-08
```

```r
if (p_value_global < 0.05) {
  cat("Le modèle est globalement significatif (p < 0.05)\n")
} else {
  cat("Le modèle n'est pas significatif (p >= 0.05)\n")
}
```

```
## Le modèle est globalement significatif (p < 0.05)
```

```r
cat("\n2. SIGNIFICATIVITÉ DES VARIABLES INDIVIDUELLES\n")
```

```
##
## 2. SIGNIFICATIVITÉ DES VARIABLES INDIVIDUELLES
```

```r
coef_summary <- summary(m2)$coefficients
for (i in 2:nrow(coef_summary)) {
  var_name <- rownames(coef_summary)[i]
  p_val <- coef_summary[i, 4]
  cat("   ", var_name, ":")
  if (p_val < 0.05) {
    cat(" SIGNIFICATIF (p =", format.pval(p_val), ")\n")
  } else {
    cat(" NON significatif (p =", format.pval(p_val), ")\n")
  }
}
```

```
##     X1 : SIGNIFICATIF (p = 0.0090533 )
##     X4 : SIGNIFICATIF (p = 0.0052493 )
##     X6 : SIGNIFICATIF (p = 8.0738e-06 )
```

Contrariément au modèle complet, les variables individuelles sont cette fois significatives.

4-B Modèle avec 2 variables

```r
variables_selectionnees <- names(which(res$which[2, ]))
variables_selectionnees <- variables_selectionnees[variables_selectionnees != "(Intercept)"]
print(variables_selectionnees)
```

```
## [1] "X6" "X7"
```

5 Changement de critère

Critère Cp de Mallows

```r
cat("\n=== MEILLEUR MODÈLE SELON LE CRITÈRE Cp ===\n\n")
```

```
##
## === MEILLEUR MODÈLE SELON LE CRITÈRE Cp ===
```

```r
# Identifier le modèle avec le Cp minimum
cp_values <- res$cp
meilleur_idx_cp <- which.min(cp_values)
cat("Nombre de variables du meilleur modèle :", meilleur_idx_cp, "\n")
```

```
## Nombre de variables du meilleur modèle : 3
```

```r
cat("Valeur du Cp minimum :", round(cp_values[meilleur_idx_cp], 4), "\n\n")
```

```
## Valeur du Cp minimum : 0.5752
```

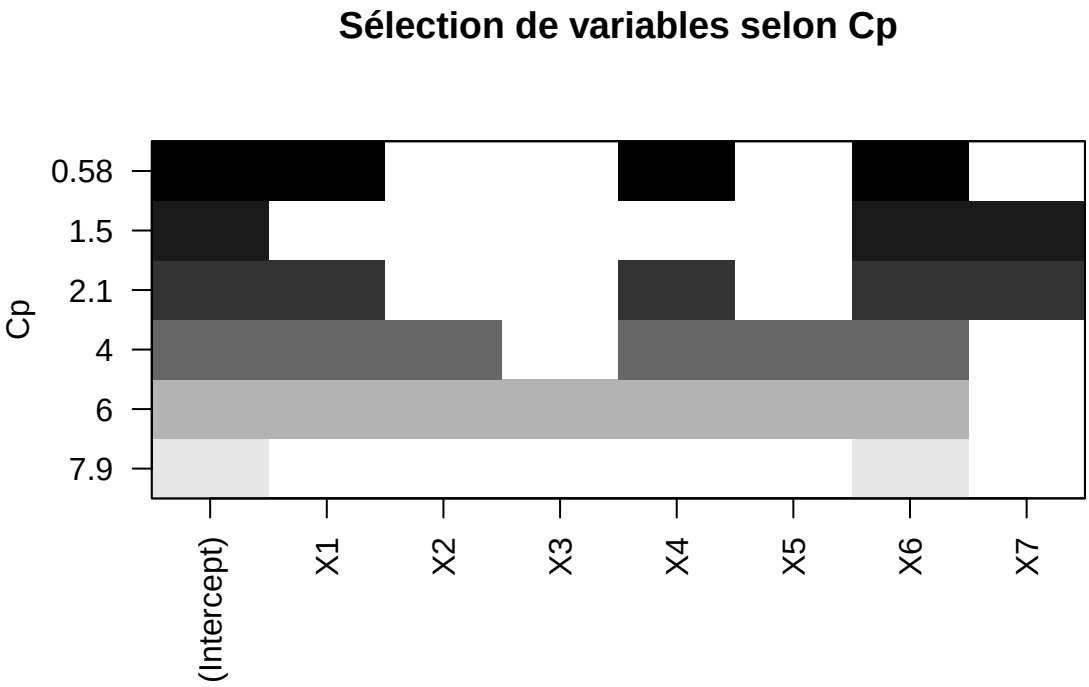
```r
# Afficher les variables sélectionnées
cat("Variables incluses dans le meilleur modèle :\n")
```

```
## Variables incluses dans le meilleur modèle :
```

```r
variables_cp <- names(which(res$which[meilleur_idx_cp, ]))
variables_cp <- variables_cp[variables_cp != "(Intercept)"]
print(variables_cp)
```

```
## [1] "X1" "X4" "X6"
```

```r
# Plot spécifique pour Cp
plot(choix, scale = "Cp", main = "Sélection de variables selon Cp")
```



Réponse Cp : Le meilleur modèle selon le critère Cp de Mallows contient **3 variables** : X1, X4, X6.

Critère R^2 ajusté

```
cat("\n=== MEILLEUR MODÈLE SELON LE CRITÈRE R² AJUSTÉ ===\n\n")
```

```
##
## === MEILLEUR MODÈLE SELON LE CRITÈRE R² AJUSTÉ ===
```

```
# Identifier le modèle avec le R² ajusté maximum
adjr2_values <- res$adjr2
meilleur_idx_adjr2 <- which.max(adjr2_values)
cat("Nombre de variables du meilleur modèle :", meilleur_idx_adjr2, "\n")
```

```
## Nombre de variables du meilleur modèle : 3
```

```
cat("Valeur du R² ajusté maximum :", round(adjr2_values[meilleur_idx_adjr2], 4), "\n\n")
```

```
## Valeur du R² ajusté maximum : 0.9882
```

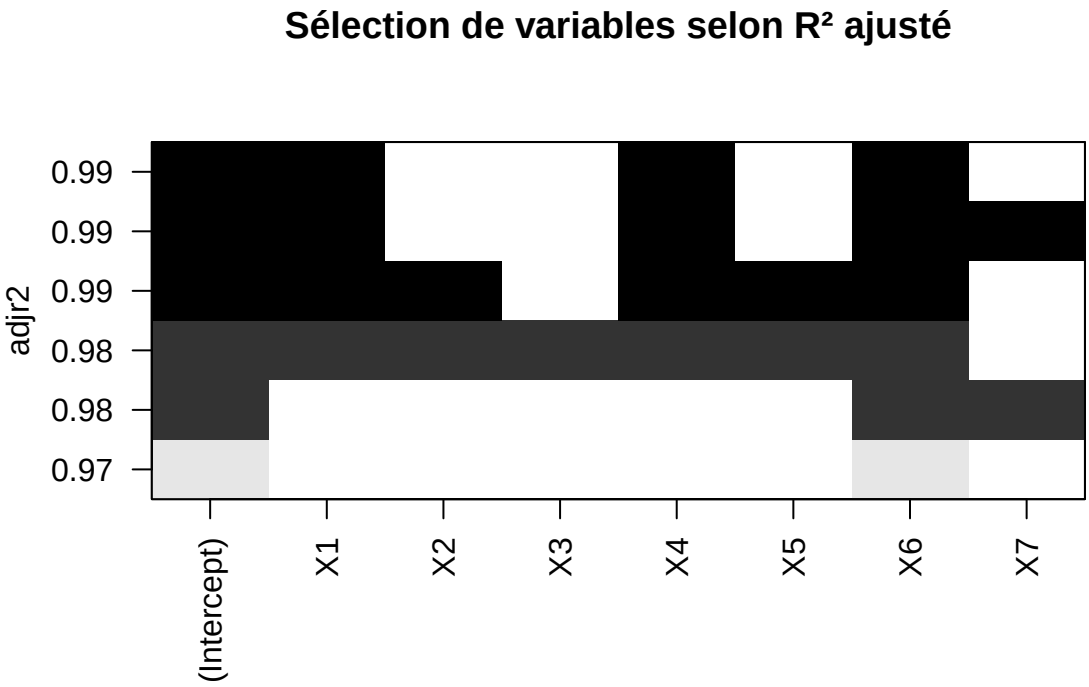
```
# Afficher les variables sélectionnées
cat("Variables incluses dans le meilleur modèle :\n")
```

```
## Variables incluses dans le meilleur modèle :
```

```
variables_adjr2 <- names(which(res$which[meilleur_idx_adjr2, ]))
variables_adjr2 <- variables_adjr2[variables_adjr2 != "(Intercept)"]
print(variables_adjr2)
```

```
## [1] "X1" "X4" "X6"
```

```
# Plot spécifique pour R² ajusté
plot(choix, scale = "adjr2", main = "Sélection de variables selon R² ajusté")
```



Réponse R² ajusté : Le meilleur modèle selon le critère R² ajusté contient **3 variables** : X1, X4, X6.

Critère R²

```
cat("\n=== MEILLEUR MODÈLE SELON LE CRITÈRE R² ===\n\n")
```

```
##
## === MEILLEUR MODÈLE SELON LE CRITÈRE R² ===
```

```
# Identifier le modèle avec le R² maximum
r2_values <- res$rsq
meilleur_idx_r2 <- which.max(r2_values)
cat("Nombre de variables du meilleur modèle :", meilleur_idx_r2, "\n")
```

```
## Nombre de variables du meilleur modèle : 6
```

```
cat("Valeur du R² maximum :", round(r2_values[meilleur_idx_r2], 4), "\n\n")
```

```
## Valeur du R² maximum : 0.9925
```

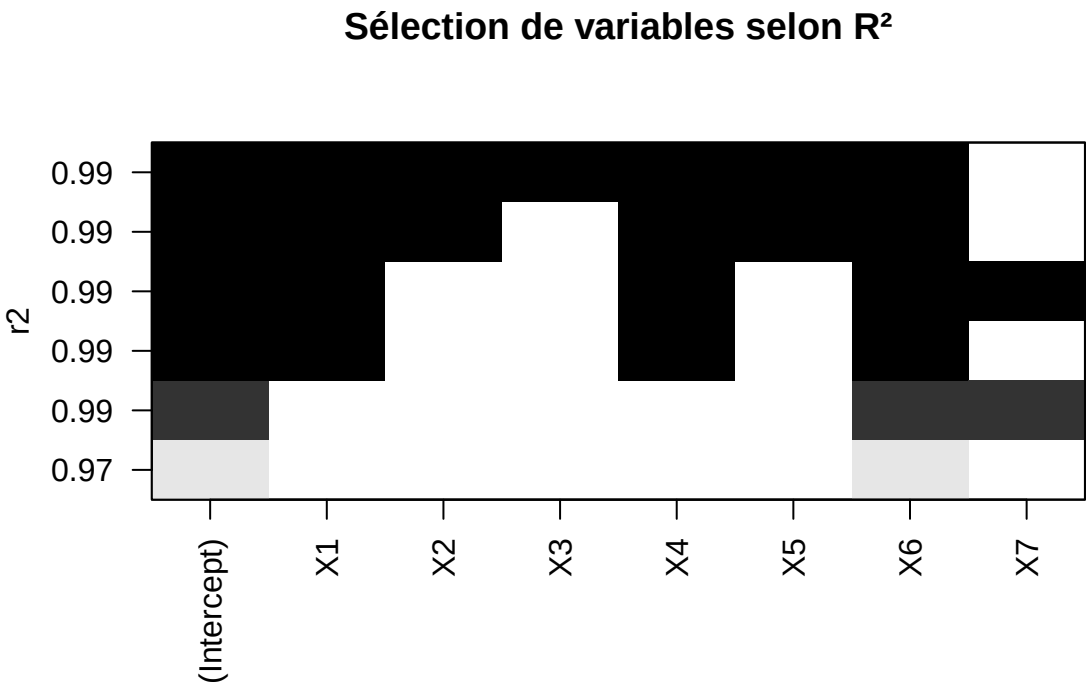
```
# Afficher les variables sélectionnées
cat("Variables incluses dans le meilleur modèle :\n")
```

```
## Variables incluses dans le meilleur modèle :
```

```
variables_r2 <- names(which(res$which[meilleur_idx_r2, ]))
variables_r2 <- variables_r2[variables_r2 != "(Intercept)"]
print(variables_r2)
```

```
## [1] "X1" "X2" "X3" "X4" "X5" "X6"
```

```
# Plot spécifique pour R²
plot(choix, scale = "r2", main = "Sélection de variables selon R²")
```



Réponse R² : Le meilleur modèle selon le critère R^2 contient **6 variables** : X1, X2, X3, X4, X5, X6.

Synthèse comparative

```r
cat("\n=== SYNTHÈSE DES CRITÈRES DE SÉLECTION ===\n\n")
```

```
##
## === SYNTHÈSE DES CRITÈRES DE SÉLECTION ===
```

```r
synthese <- data.frame(
  Critere = c("BIC", "Cp de Mallows", "R² ajusté", "R²"),
  Nombre_Variables = c(
    meilleur_idx,
    meilleur_idx_cp,
    meilleur_idx_adjr2,
    meilleur_idx_r2
  ),
  Variables = c(
    paste(variables_selectionnees, collapse = ", "),
    paste(variables_cp, collapse = ", "),
    paste(variables_adjr2, collapse = ", "),
    paste(variables_r2, collapse = ", ")
  ),
  Valeur = c(
    round(res$bic[meilleur_idx], 4),
    round(res$cp[meilleur_idx_cp], 4),
    round(res$adjr2[meilleur_idx_adjr2], 4),
    round(res$rsq[meilleur_idx_r2], 4)
  )
)

knitr::kable(
  synthese,
  col.names = c("Critère", "Nb Variables", "Variables sélectionnées", "Valeur")
)
```

Critère	Nb Variables	Variables sélectionnées	Valeur
BIC	3	X6, X7	-47.2047
Cp de Mallows	3	X1, X4, X6	0.5752
R^2 ajusté	3	X1, X4, X6	0.9882
R^2	6	X1, X2, X3, X4, X5, X6	0.9925

6 Sélection de variables pas-à-pas

Les recherches précédentes étaient exhaustives. Cela pose un problème lorsque le nombre de variables est grand. Faisons une sélection de variables pas-à-pas.

Définition de la fonction PRESS

```r
press <- function(fit) {
  h <- lm.influence(fit)$hat
  sqrt(mean((residuals(fit) / (1 - h))^2))
}
```

Sélection backward (pas-à-pas descendante)

```
library(MASS)

# Modèle sans aucune variable explicative
m_0 <- lm(Y ~ 1, data = data)

# Modèle avec toutes les variables
m_all <- lm(Y ~ ., data = data)

# Sélection backward (on part du modèle complet et on retire des variables)
cat("=== SÉLECTION BACKWARD ===\n\n")
```

```
## === SÉLECTION BACKWARD ===
```

```
m_back <- stepAIC(m_all, direction = "backward", trace = TRUE)
```

```
## Start:  AIC=-0.8
## Y ~ X1 + X2 + X3 + X4 + X5 + X6 + X7
##
##
## Step:  AIC=-0.8
## Y ~ X1 + X2 + X3 + X4 + X5 + X6
##
##        Df Sum of Sq    RSS      AIC
## - X3    1  0.002218 3.4983 -2.79158
## - X6    1  0.027082 3.5232 -2.70659
## - X5    1  0.084293 3.5804 -2.51329
## - X2    1  0.102814 3.5989 -2.45138
## - X4    1  0.135593 3.6317 -2.34258
## - X1    1  0.159001 3.6551 -2.26548
## <none>              3.4961 -0.79919
##
## Step:  AIC=-2.79
## Y ~ X1 + X2 + X4 + X5 + X6
##
##        Df Sum of Sq    RSS     AIC
## - X6    1   0.08997 3.5883 -4.4869
## - X5    1   0.23972 3.7380 -3.9962
## - X2    1   0.28702 3.7853 -3.8454
## - X4    1   0.36541 3.8637 -3.5994
## - X1    1   0.42297 3.9213 -3.4219
## <none>              3.4983 -2.7916
##
## Step:  AIC=-4.49
## Y ~ X1 + X2 + X4 + X5
##
##        Df Sum of Sq     RSS    AIC
## <none>               3.588 -4.487
## - X5    1     3.533   7.121  1.738
## - X2    1    50.419  54.008 26.051
## - X1    1    92.385  95.973 32.950
## - X4    1   135.027 138.615 37.362
```

```r
cat("\n--- Résumé du modèle backward ---\n")
```

```
##
## --- Résumé du modèle backward ---
```

```r
print(summary(m_back))
```

```
##
## Call:
## lm(formula = Y ~ X1 + X2 + X4 + X5, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.8147 -0.2742 -0.1112  0.2329  1.2051
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  100.980      0.749 134.828 3.26e-13 ***
## X1           -40.147      2.990 -13.425 2.98e-06 ***
## X2           -15.152      1.528  -9.918 2.26e-05 ***
## X4           -21.936      1.352 -16.230 8.21e-07 ***
## X5           -12.773      4.866  -2.625   0.0341 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.716 on 7 degrees of freedom
## Multiple R-squared:  0.9923, Adjusted R-squared:  0.9879
## F-statistic: 226.4 on 4 and 7 DF,  p-value: 1.768e-07
```

Sélection forward (pas-à-pas ascendante)

```r
# Sélection forward (on part du modèle vide et on ajoute des variables)
cat("\n=== SÉLECTION FORWARD ===\n\n")
```

```
##
## === SÉLECTION FORWARD ===
```

```r
m_forw <- stepAIC(
  m_0,
  direction = "forward",
  scope = list(upper = m_all, lower = m_0),
  trace = TRUE
)
```

```
## Start:  AIC=45.96
## Y ~ 1
##
##        Df Sum of Sq    RSS    AIC
## + X6    1    453.93  13.86  5.732
```

```
## + X3    1    328.47 139.32 33.423
## + X1    1    327.96 139.84 33.467
## + X7    1    256.94 210.86 38.395
## + X4    1    233.64 234.16 39.653
## + X5    1    114.07 353.73 44.604
## <none>              467.80 45.958
## + X2    1      2.35 465.45 47.897
##
## Step:  AIC=5.73
## Y ~ X6
##
##        Df Sum of Sq     RSS     AIC
## + X7    1    7.3485  6.5152 -1.3292
## + X2    1    6.7120  7.1517 -0.2106
## + X4    1    4.0095  9.8542  3.6359
## + X3    1    2.6671 11.1967  5.1685
## + X1    1    2.6534 11.2104  5.1832
## <none>              13.8638  5.7325
## + X5    1    0.8501 13.0136  6.9731
##
## Step:  AIC=-1.33
## Y ~ X6 + X7
##
##        Df Sum of Sq    RSS      AIC
## + X1    1   1.42728 5.0880 -2.29636
## + X3    1   1.38121 5.1340 -2.18821
## + X5    1   1.08694 5.4283 -1.51938
## <none>              6.5152 -1.32916
## + X4    1   0.36245 6.1528 -0.01603
## + X2    1   0.01238 6.5029  0.64802
##
## Step:  AIC=-2.3
## Y ~ X6 + X7 + X1
##
##        Df Sum of Sq    RSS     AIC
## + X4    1   1.52705 3.5609 -4.5787
## <none>              5.0880 -2.2964
## + X3    1   0.49613 4.5918 -1.5275
## + X5    1   0.44979 4.6382 -1.4071
## + X2    1   0.06527 5.0227 -0.4513
##
## Step:  AIC=-4.58
## Y ~ X6 + X7 + X1 + X4
##
##        Df Sum of Sq    RSS     AIC
## <none>              3.5609 -4.5787
## + X2    1  0.050479 3.5104 -2.7500
## + X5    1  0.049644 3.5113 -2.7472
## + X3    1  0.001742 3.5592 -2.5846
```

```
cat("\n--- Résumé du modèle forward ---\n")
```

```
##
## --- Résumé du modèle forward ---
```

```r
print(summary(m_forw))
```

```
##
## Call:
## lm(formula = Y ~ X6 + X7 + X1 + X4, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.94758 -0.32950 -0.04971  0.10991  1.11211
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   86.037      1.010  85.166  8.1e-12 ***
## X6            13.782      2.770   4.976  0.00161 **
## X7            45.069     48.573   0.928  0.38436
## X1           -22.500      9.968  -2.257  0.05857 .
## X4           -10.353      5.976  -1.733  0.12677
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7132 on 7 degrees of freedom
## Multiple R-squared:  0.9924, Adjusted R-squared:  0.988
## F-statistic: 228.1 on 4 and 7 DF,  p-value: 1.721e-07
```

Sélection stepwise (bidirectionnelle)

```r
# Sélection stepwise (on peut ajouter ou retirer des variables)
cat("\n=== SÉLECTION STEPWISE (BIDIRECTIONNELLE) ===\n\n")
```

```
##
## === SÉLECTION STEPWISE (BIDIRECTIONNELLE) ===
```

```r
m_stepwise <- stepAIC(
  m_0,
  direction = "both",
  scope = list(upper = m_all, lower = m_0),
  trace = TRUE
)
```

```
## Start:  AIC=45.96
## Y ~ 1
##
##        Df Sum of Sq    RSS    AIC
## + X6    1    453.93  13.86  5.732
## + X3    1    328.47 139.32 33.423
## + X1    1    327.96 139.84 33.467
## + X7    1    256.94 210.86 38.395
## + X4    1    233.64 234.16 39.653
## + X5    1    114.07 353.73 44.604
## <none>              467.80 45.958
```

```
## + X2    1      2.35 465.45 47.897
##
## Step:  AIC=5.73
## Y ~ X6
##
##        Df Sum of Sq    RSS    AIC
## + X7    1      7.35   6.52 -1.329
## + X2    1      6.71   7.15 -0.211
## + X4    1      4.01   9.85  3.636
## + X3    1      2.67  11.20  5.169
## + X1    1      2.65  11.21  5.183
## <none>               13.86  5.732
## + X5    1      0.85  13.01  6.973
## - X6    1    453.93 467.80 45.958
##
## Step:  AIC=-1.33
## Y ~ X6 + X7
##
##        Df Sum of Sq     RSS    AIC
## + X1    1     1.427   5.088 -2.296
## + X3    1     1.381   5.134 -2.188
## + X5    1     1.087   5.428 -1.519
## <none>                6.515 -1.329
## + X4    1     0.362   6.153 -0.016
## + X2    1     0.012   6.503  0.648
## - X7    1     7.349  13.864  5.732
## - X6    1   204.343 210.858 38.395
##
## Step:  AIC=-2.3
## Y ~ X6 + X7 + X1
##
##        Df Sum of Sq    RSS     AIC
## + X4    1     1.527  3.561 -4.5787
## <none>               5.088 -2.2964
## + X3    1     0.496  4.592 -1.5275
## + X5    1     0.450  4.638 -1.4071
## - X1    1     1.427  6.515 -1.3292
## + X2    1     0.065  5.023 -0.4513
## - X7    1     6.122 11.210  5.1832
## - X6    1    93.651 98.739 31.2909
##
## Step:  AIC=-4.58
## Y ~ X6 + X7 + X1 + X4
##
##        Df Sum of Sq     RSS     AIC
## - X7    1    0.4379  3.9988 -5.1868
## <none>               3.5609 -4.5787
## + X2    1    0.0505  3.5104 -2.7500
## + X5    1    0.0496  3.5113 -2.7472
## + X3    1    0.0017  3.5592 -2.5846
## - X4    1    1.5271  5.0880 -2.2964
## - X1    1    2.5919  6.1528 -0.0160
## - X6    1   12.5943 16.1552 11.5680
##
```

```
## Step:  AIC=-5.19
## Y ~ X6 + X1 + X4
##
##        Df Sum of Sq    RSS     AIC
## <none>               3.999 -5.1868
## + X7    1     0.438  3.561 -4.5787
## + X2    1     0.261  3.738 -3.9962
## + X5    1     0.214  3.785 -3.8454
## + X3    1     0.177  3.821 -3.7316
## - X1    1     5.855  9.854  3.6359
## - X4    1     7.212 11.210  5.1832
## - X6    1    50.660 54.659 24.1945
```

```
cat("\n--- Résumé du modèle stepwise ---\n")
```

```
##
## --- Résumé du modèle stepwise ---
```

```
print(summary(m_stepwise))
```

```
##
## Call:
## lm(formula = Y ~ X6 + X1 + X4, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.00154 -0.41198  0.02205  0.29286  1.00148
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  85.9435     0.9964  86.255 3.64e-13 ***
## X6           15.8852     1.5779  10.067 8.07e-06 ***
## X1          -14.0924     4.1175  -3.423  0.00905 **
## X4           -4.9445     1.3018  -3.798  0.00525 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.707 on 8 degrees of freedom
## Multiple R-squared:  0.9915, Adjusted R-squared:  0.9882
## F-statistic: 309.3 on 3 and 8 DF,  p-value: 1.31e-08
```

Comparaison des modèles selon le critère PRESS

```
cat("\n=== COMPARAISON DES MODÈLES SELON LE CRITÈRE PRESS ===\n\n")
```

```
##
## === COMPARAISON DES MODÈLES SELON LE CRITÈRE PRESS ===
```

```r
# Calcul du PRESS pour chaque modèle
press_back <- press(m_back)
press_forw <- press(m_forw)
press_stepwise <- press(m_stepwise)

# Extraction des variables de chaque modèle
vars_back <- names(coef(m_back))[-1]  # Exclure l'intercept
vars_forw <- names(coef(m_forw))[-1]
vars_stepwise <- names(coef(m_stepwise))[-1]

# Tableau comparatif
comparaison_modeles <- data.frame(
  Methode = c("Backward", "Forward", "Stepwise"),
  Nb_Variables = c(
    length(vars_back),
    length(vars_forw),
    length(vars_stepwise)
  ),
  Variables = c(
    paste(vars_back, collapse = ", "),
    paste(vars_forw, collapse = ", "),
    paste(vars_stepwise, collapse = ", ")
  ),
  AIC = c(
    AIC(m_back),
    AIC(m_forw),
    AIC(m_stepwise)
  ),
  PRESS = c(
    press_back,
    press_forw,
    press_stepwise
  ),
  R2 = c(
    summary(m_back)$r.squared,
    summary(m_forw)$r.squared,
    summary(m_stepwise)$r.squared
  ),
  R2_ajuste = c(
    summary(m_back)$adj.r.squared,
    summary(m_forw)$adj.r.squared,
    summary(m_stepwise)$adj.r.squared
  )
)

knitr::kable(
  comparaison_modeles,
  col.names = c(
    "Méthode",
    "Nb Var",
    "Variables sélectionnées",
    "AIC",
    "PRESS",
```

```
    "R²",
    "R² ajusté"
  ),
  digits = 4
)
```

Méthode	Nb Var	Variables sélectionnées	AIC	PRESS	R^2	R^2 ajusté
Backward	4	X1, X2, X4, X5	31.5677	1.2871	0.9923	0.9879
Forward	4	X6, X7, X1, X4	31.4758	1.0121	0.9924	0.9880
Stepwise	3	X6, X1, X4	30.8677	1.0176	0.9915	0.9882

```
# Identifier le meilleur modèle selon PRESS
meilleur_press <- which.min(comparaison_modeles$PRESS)
cat("\n=== MEILLEUR MODÈLE SELON LE CRITÈRE PRESS ===\n")
```

```
##
## === MEILLEUR MODÈLE SELON LE CRITÈRE PRESS ===
```

```
cat("Méthode :", comparaison_modeles$Methode[meilleur_press], "\n")
```

```
## Méthode : Forward
```

```
cat("PRESS :", round(comparaison_modeles$PRESS[meilleur_press], 4), "\n")
```

```
## PRESS : 1.0121
```

```
cat("Variables :", comparaison_modeles$Variables[meilleur_press], "\n")
```

```
## Variables : X6, X7, X1, X4
```