½ JOURNÉE DATA SCIENCE

ÉTUDE COMPARATIVE DES MÉTHODES DE SÉLECTION DE VARIABLES DANS LE MODÈLE LINÉAIRE

- Objectif : comparer les performances de plusieurs méthodes de sélection de variables
- Méthodes : test de Student, recherche exhaustive, stepwise
- Évaluation : précision, sensibilité, spécificité, RMSE, erreur de prédiction

PROTOCOLE DE GÉNÉRATION DES DONNES

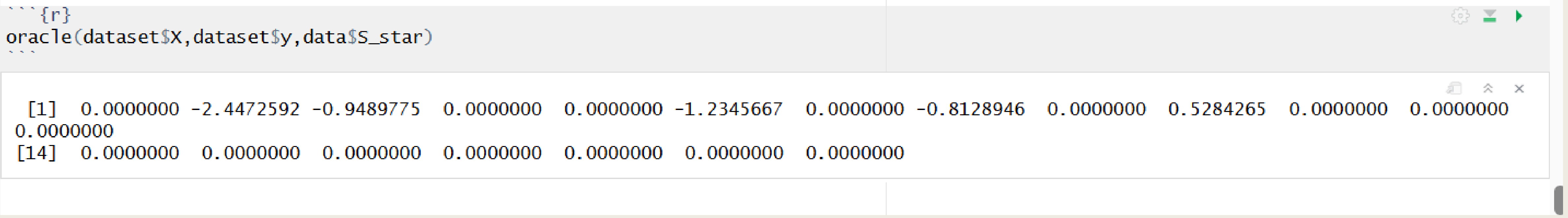
- Paramètres : n.train = 140, n.test = 60, p = 20, p_0 = 5
- Deux cas : variables indépendantes / corrélées (rho=0.6)

ESTIMATEUR ORACLE

$$\hat{\beta}_{S^*} = (X_{S^*}^T X_{S^*})^{-1} X_{S^*}^T Y$$

$$\hat{\beta}_j = 0 \quad \text{si } j \notin S^*$$

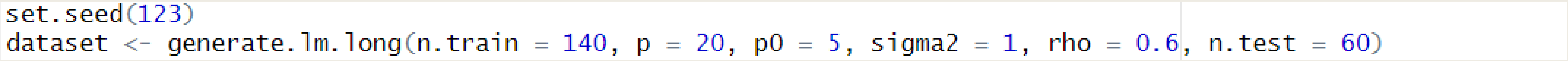
- L’oracle est une référence théorique, il connaît le vrai support (les variables réellement non nulles).
- On applique les moindres carrés ordinaires uniquement sur ces variables.
- Cela donne la performance “idéale” que les autres méthodes ne peuvent qu’approcher.



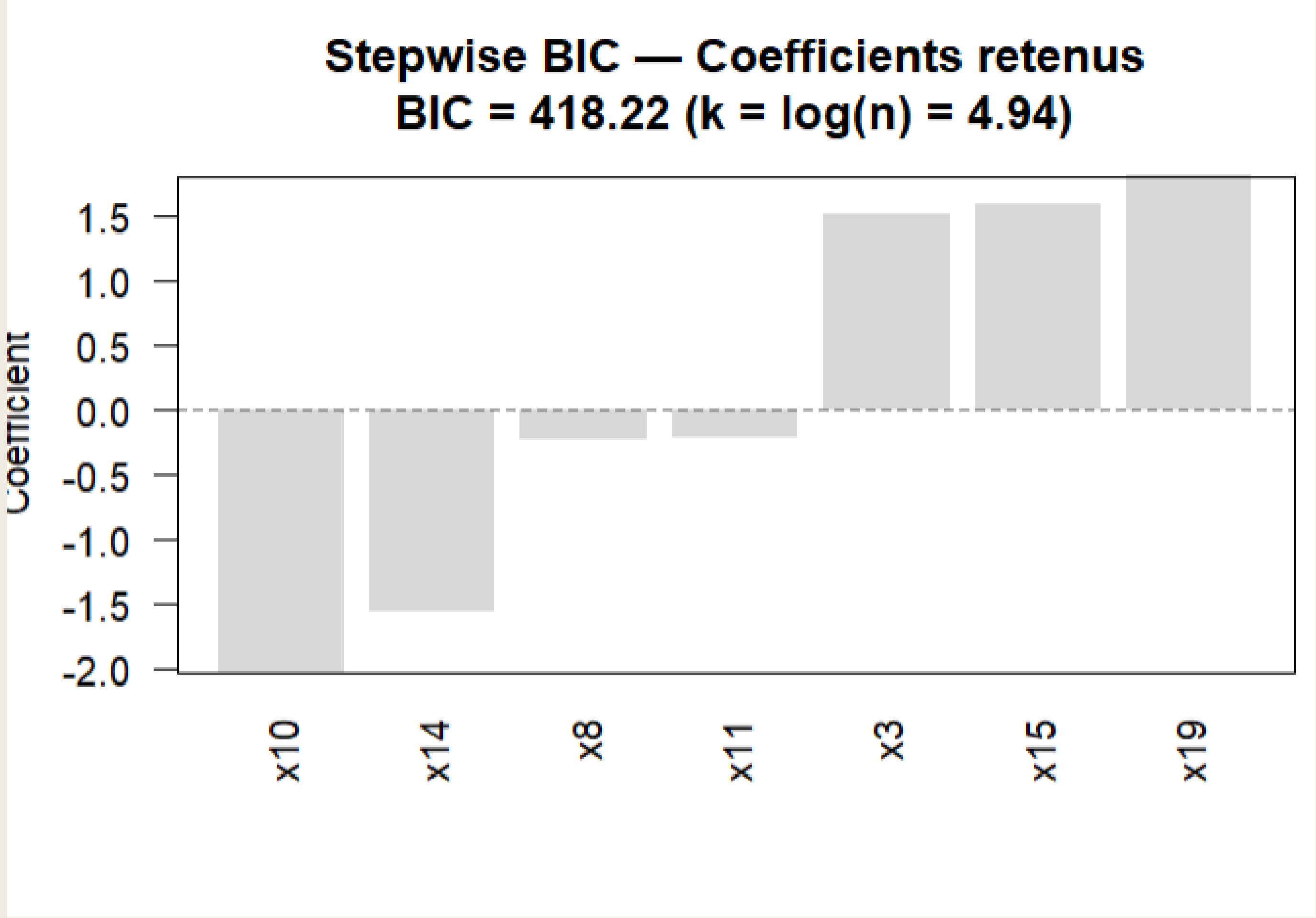
~~~
```{r}
oracle(dataset$X,dataset$y,data$S_star)
```
~~~

```
 [1]  0.0000000 -2.4472592 -0.9489775  0.0000000  0.0000000 -1.2345667  0.0000000 -0.8128946  0.0000000  0.5284265  0.0000000  0.0000000
0.0000000
[14]  0.0000000  0.0000000  0.0000000  0.0000000  0.0000000  0.0000000  0.0000000
```

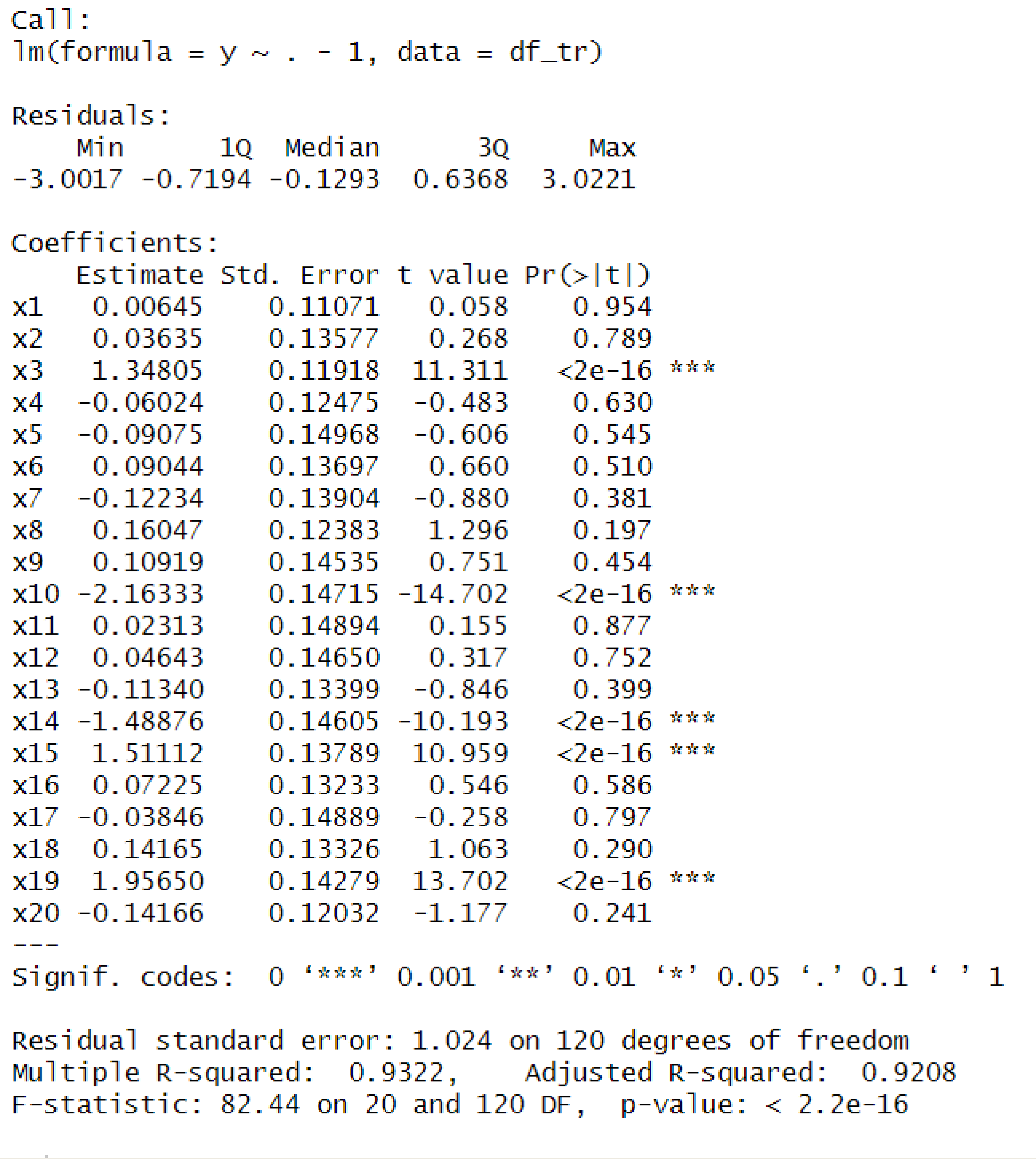
TEST DE STUDENT



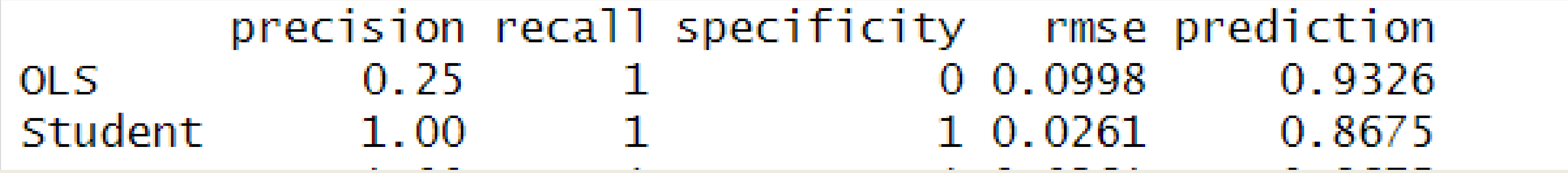
```
set.seed(123)
dataset <- generate.lm.long(n.train = 140, p = 20, p0 = 5, sigma2 = 1, rho = 0.6, n.test = 60)
```



Stepwise BIC — Coefficients retenus
BIC = 418.22 (k = log(n) = 4.94)

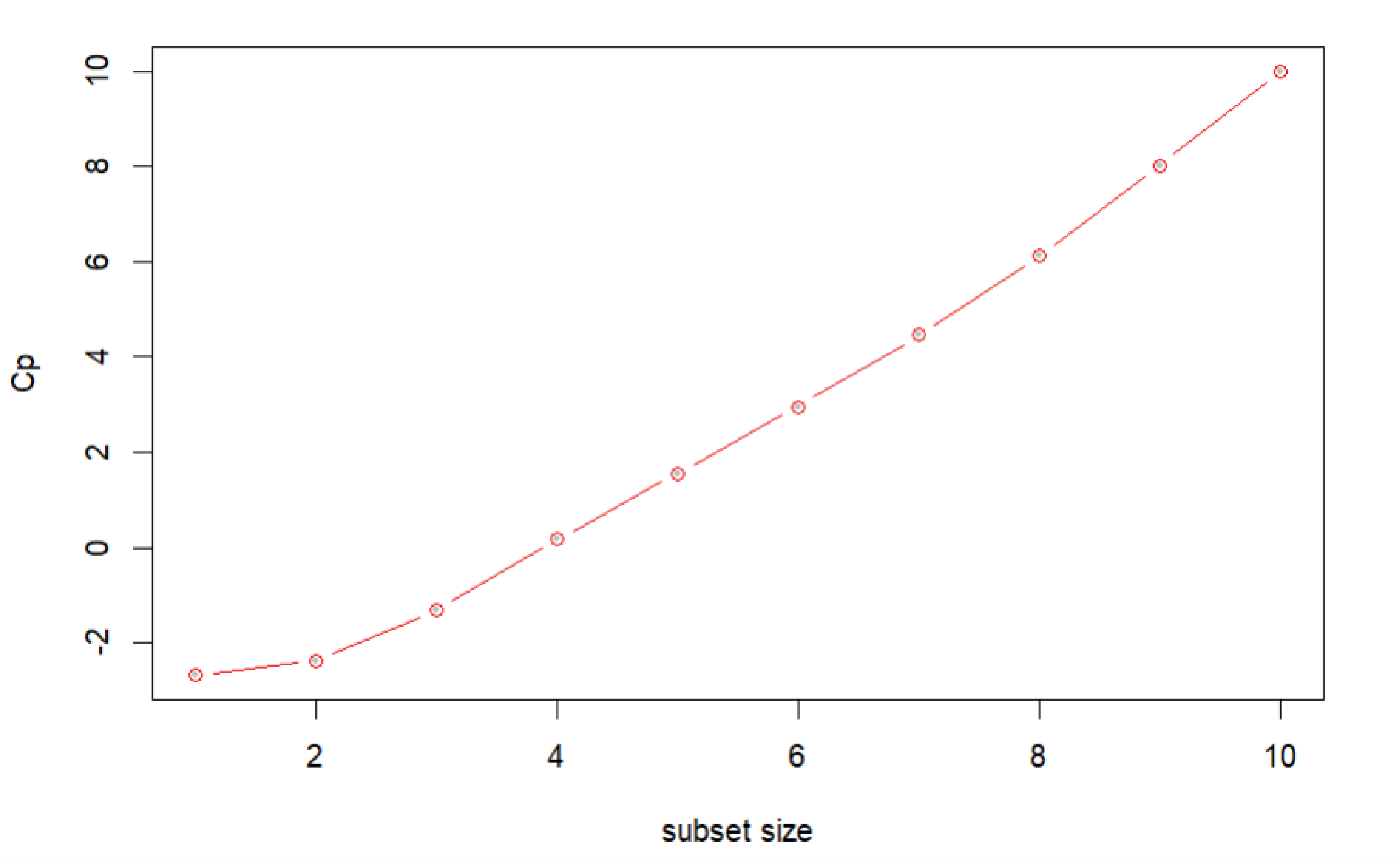


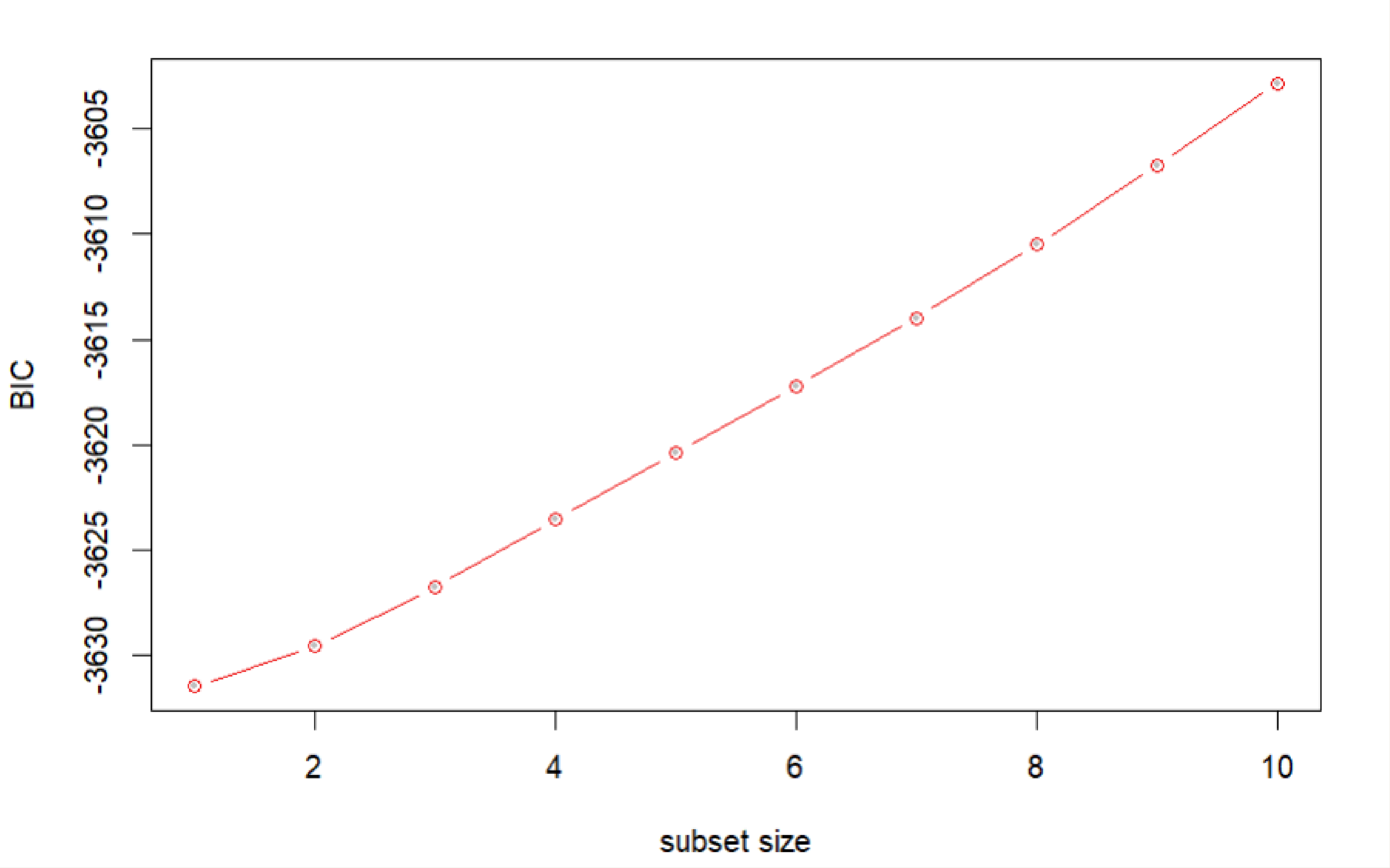
```
Call:
lm(formula = y ~ . - 1, data = df_tr)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.0017 -0.7194 -0.1293  0.6368  3.0221 

Coefficients:
    Estimate Std. Error t value Pr(>|t|)    
x1   0.00645    0.11071   0.058    0.954    
x2   0.03635    0.13577   0.268    0.789    
x3   1.34805    0.11918  11.311   <2e-16 ***
x4  -0.06024    0.12475  -0.483    0.630    
x5  -0.09075    0.14968  -0.606    0.545    
x6   0.09044    0.13697   0.660    0.510    
x7  -0.12234    0.13904  -0.880    0.381    
x8   0.16047    0.12383   1.296    0.197    
x9   0.10919    0.14535   0.751    0.454    
x10 -2.16333    0.14715 -14.702   <2e-16 ***
x11  0.02313    0.14894   0.155    0.877    
x12  0.04643    0.14650   0.317    0.752    
x13 -0.11340    0.13399  -0.846    0.399    
x14 -1.48876    0.14605 -10.193   <2e-16 ***
x15  1.51112    0.13789  10.959   <2e-16 ***
x16  0.07225    0.13233   0.546    0.586    
x17 -0.03846    0.14889  -0.258    0.797    
x18  0.14165    0.13326   1.063    0.290    
x19  1.95650    0.14279  13.702   <2e-16 ***
x20 -0.14166    0.12032  -1.177    0.241    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.024 on 120 degrees of freedom
Multiple R-squared:  0.9322,	Adjusted R-squared:  0.9208 
F-statistic: 82.44 on 20 and 120 DF,  p-value: < 2.2e-16
```



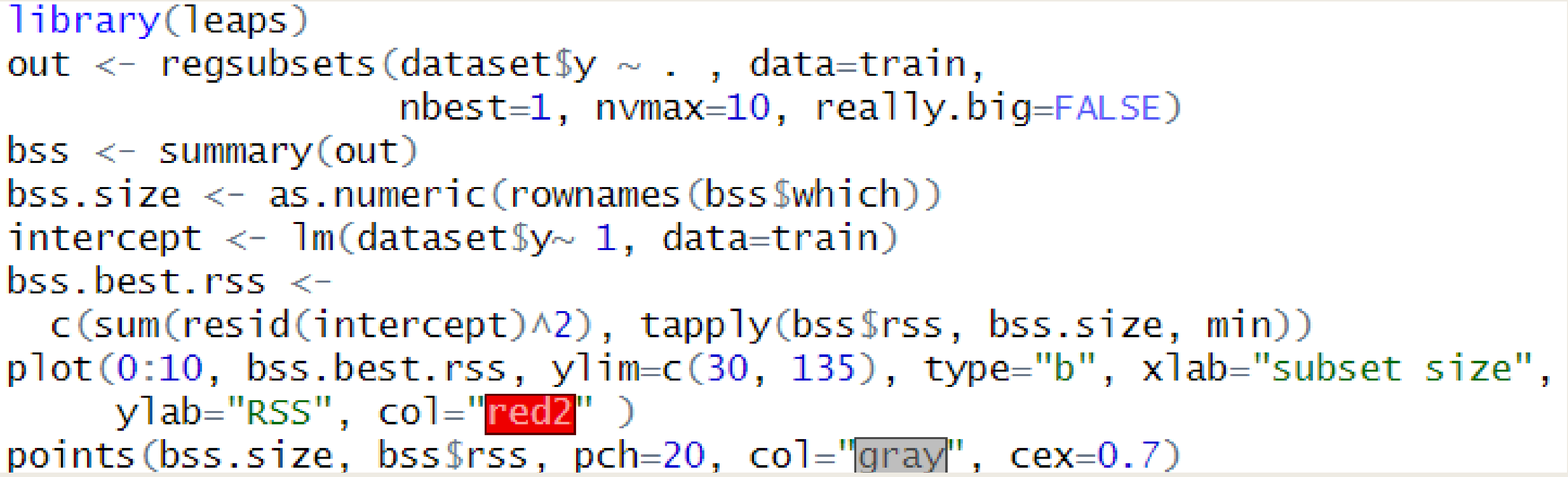
```
        precision recall specificity   rmse prediction
OLS          0.25      1           0 0.0998     0.9326
Student      1.00      1           1 0.0261     0.8675
```

MÉTHODE RECHERCHE EXHAUSTIVE MODÈLE LINÉAIRE

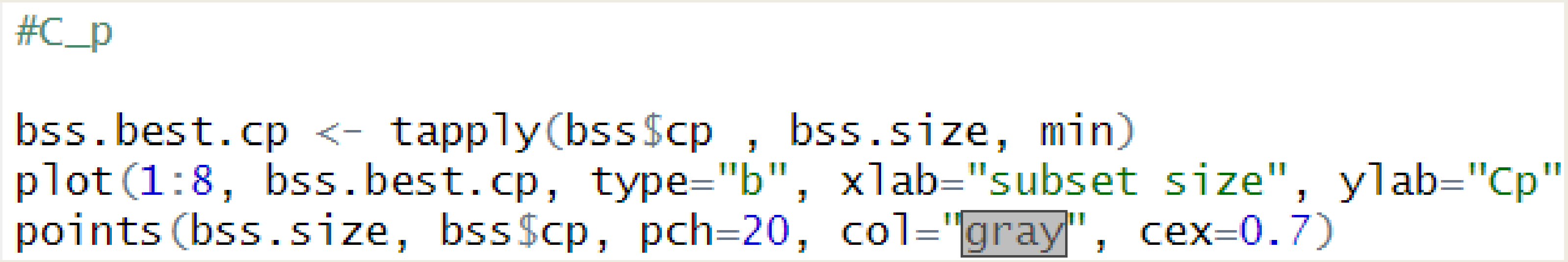




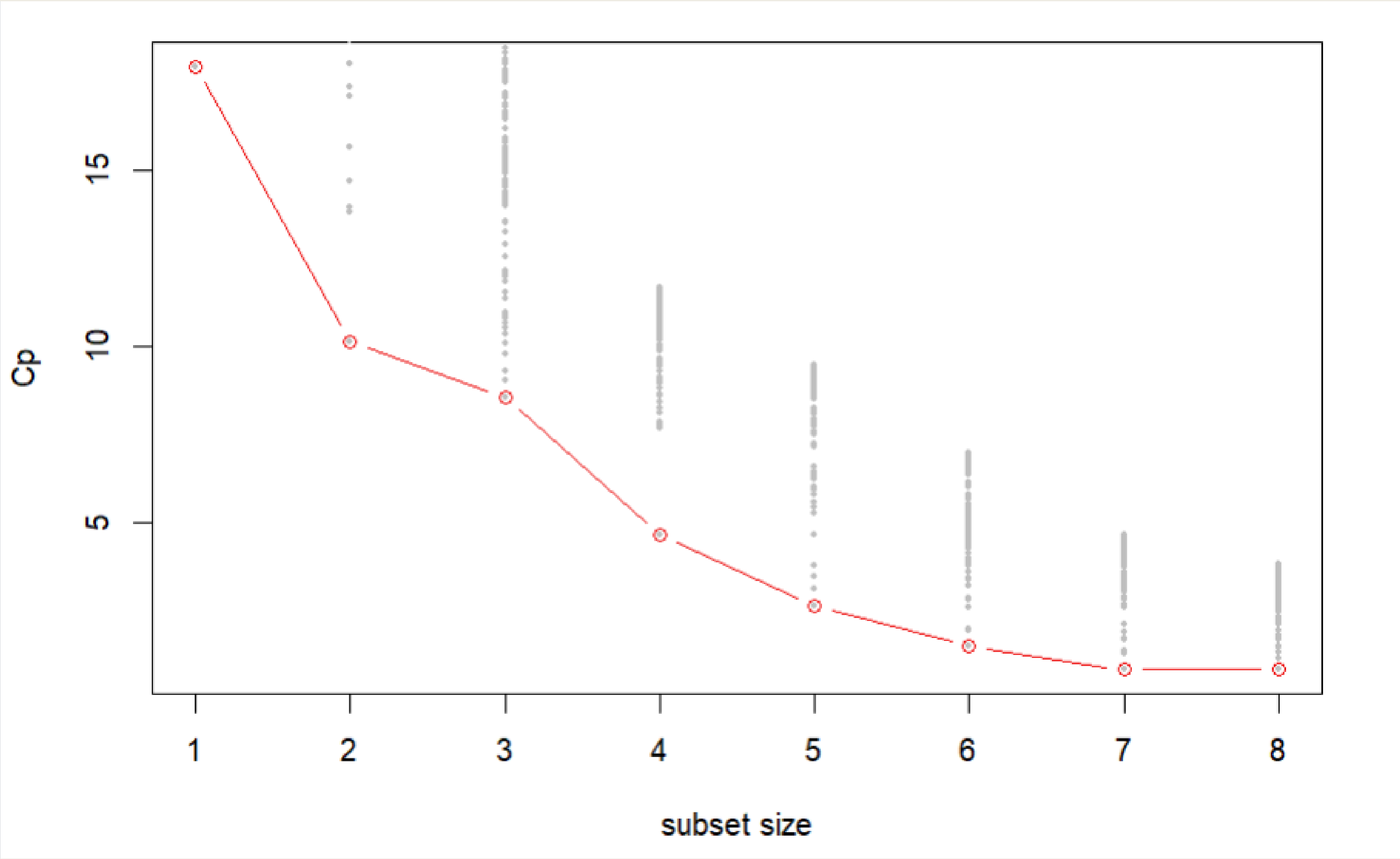
MÉTHODE RECHERCHE EXHAUSTIVE



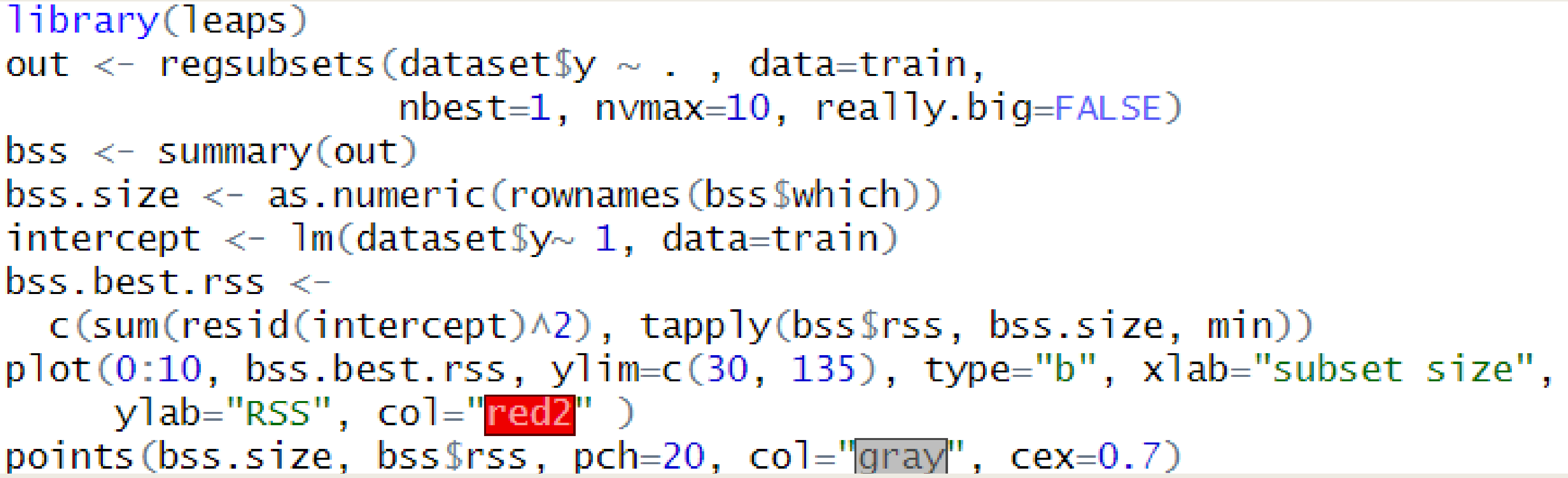
```
library(leaps)
out <- regsubsets(dataset$y ~ . , data=train,
                  nbest=1, nvmax=10, really.big=FALSE)
bss <- summary(out)
bss.size <- as.numeric(rownames(bss$which))
intercept <- lm(dataset$y~ 1, data=train)
bss.best.rss <-
  c(sum(resid(intercept)^2), tapply(bss$rss, bss.size, min))
plot(0:10, bss.best.rss, ylim=c(30, 135), type="b", xlab="subset size",
     ylab="RSS", col="red2" )
points(bss.size, bss$rss, pch=20, col="gray", cex=0.7)
```



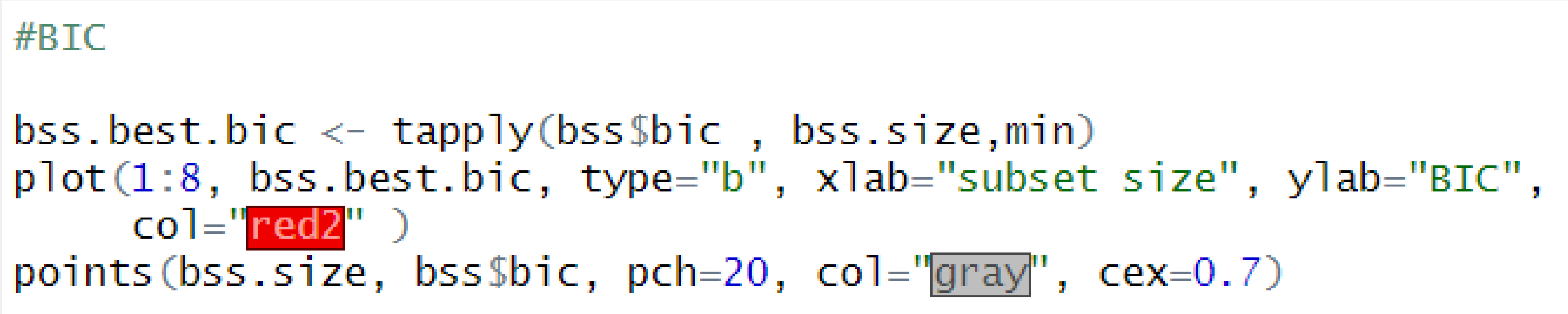
```
#C_p

bss.best.cp <- tapply(bss$cp , bss.size, min)
plot(1:8, bss.best.cp, type="b", xlab="subset size", ylab="Cp"
points(bss.size, bss$cp, pch=20, col="gray", cex=0.7)
```



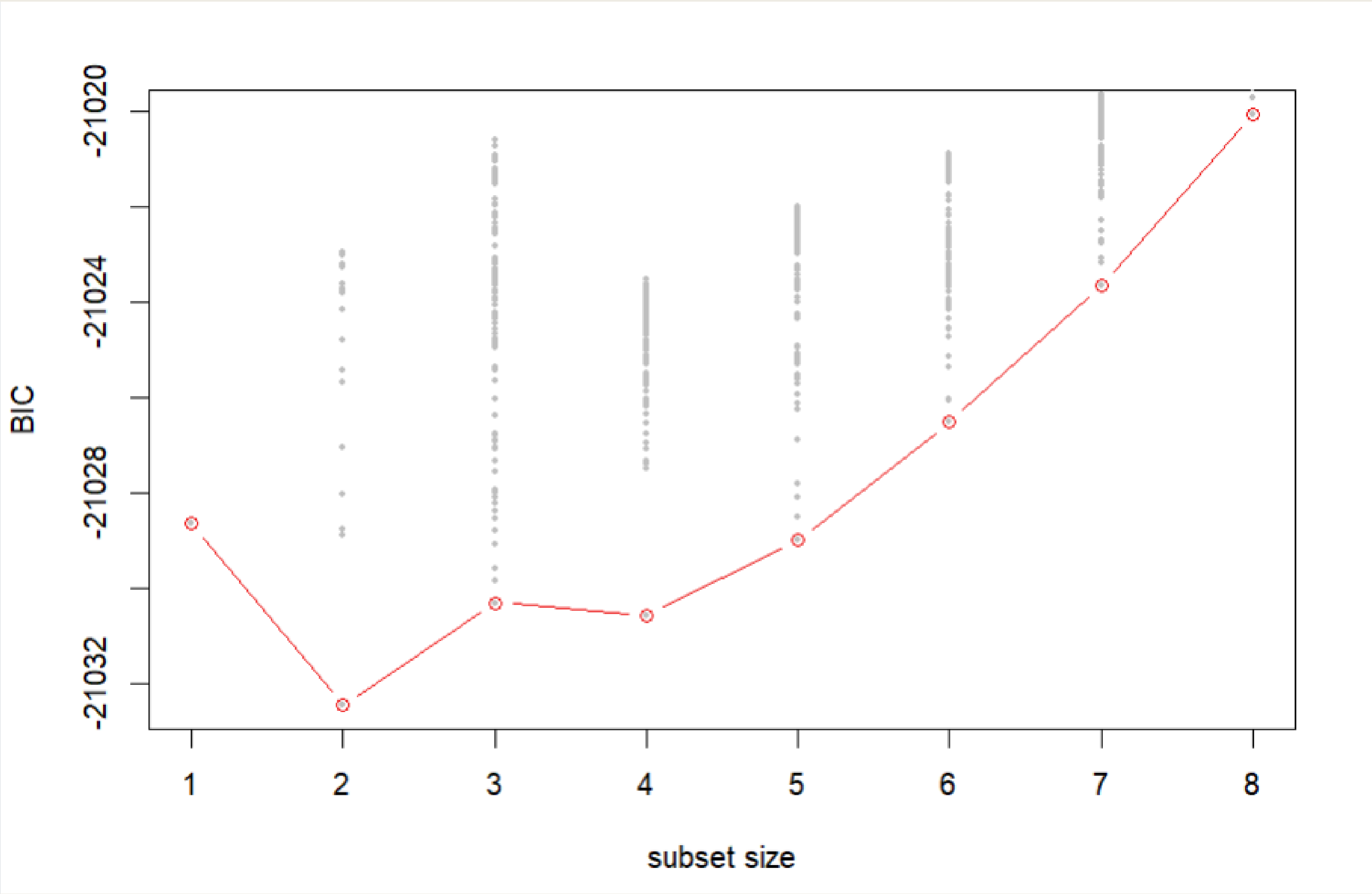
MÉTHODE RECHERCHE EXHAUSTIVE



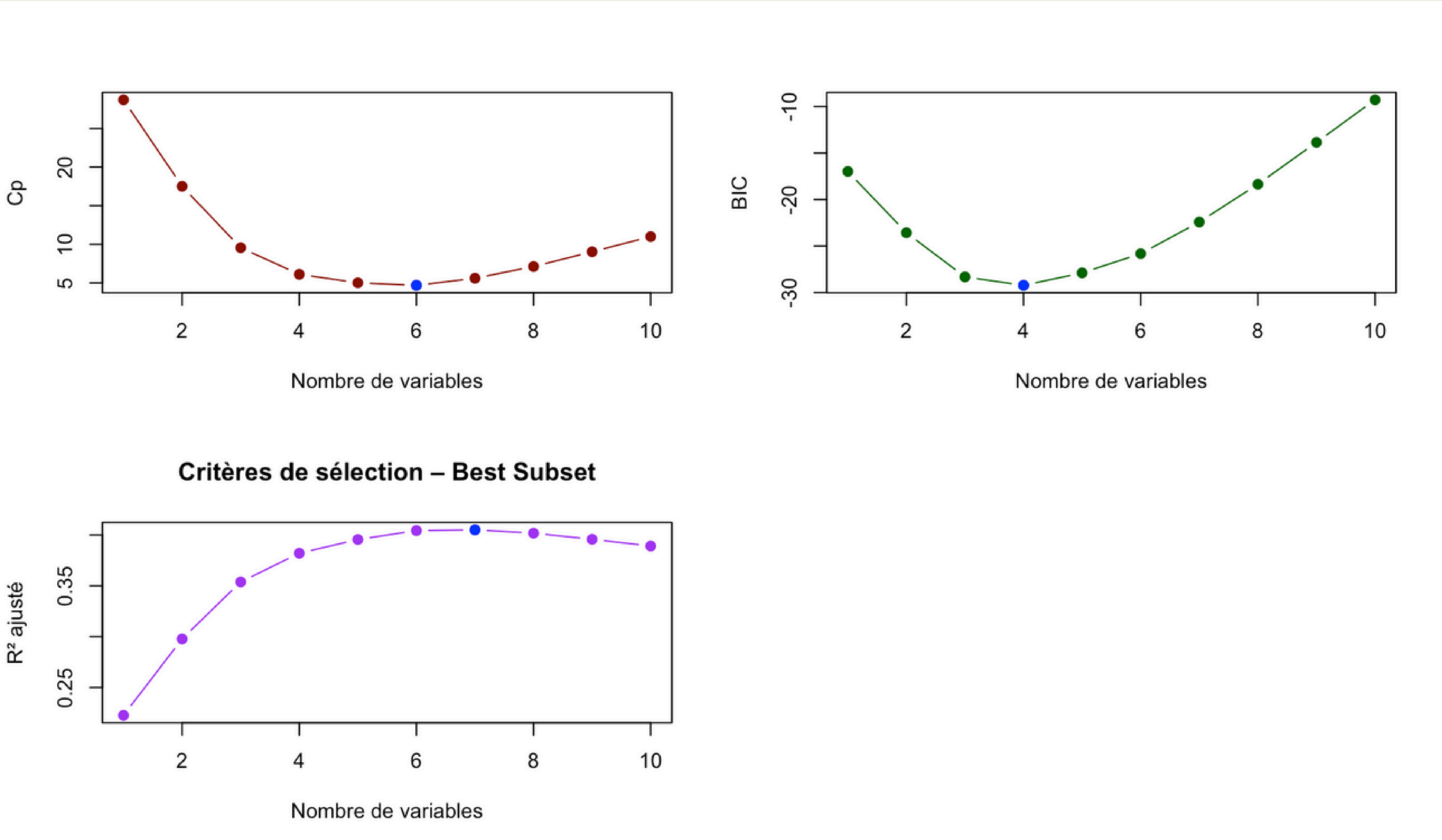
```
library(leaps)
out <- regsubsets(dataset$y ~ . , data=train,
                  nbest=1, nvmax=10, really.big=FALSE)
bss <- summary(out)
bss.size <- as.numeric(rownames(bss$which))
intercept <- lm(dataset$y~ 1, data=train)
bss.best.rss <-
  c(sum(resid(intercept)^2), tapply(bss$rss, bss.size, min))
plot(0:10, bss.best.rss, ylim=c(30, 135), type="b", xlab="subset size",
     ylab="RSS", col="red2" )
points(bss.size, bss$rss, pch=20, col="gray", cex=0.7)
```



```
#BIC

bss.best.bic <- tapply(bss$bic , bss.size,min)
plot(1:8, bss.best.bic, type="b", xlab="subset size", ylab="BIC",
     col="red2" )
points(bss.size, bss$bic, pch=20, col="gray", cex=0.7)
```



SÉLECTION DE VARIABLES



Variable	β_vrai	β_AIC	β_BIC	β_BEST
X2	-1.46	-0.64	-0.77	-0.77
X3	1.89	1.27	1.28	1.28
X8	1.96	1.96	2.00	2.00
X10	1.55	1.26	1.21	1.21

CONCLUSION ET PERSPECTIVES

Points :

- Protocole de simulation validé
- Stepwise BIC le plus stable
- Pistes : Lasso, Elastic Net, robustesse aux écarts