Régression linéaire

Eric Marcon

21 novembre 2018

Résumé

L'objectif du cours est de comprendre les facteurs qui influent sur la précision de l'estimation d'une régression linéaire, pour permettre de choisir un design expérimental adapté à la question posée.

Table des matières

1 Contexte

On estime l'effet de deux variables explicatives sur une variable expliquée dans un modèle de régression linéaire classique. Les données sont manipulées pour mettre en évidence les effets du design expérimental.

Le modèle est le suivant :

$$y = a_1 x_1 + a_2 x_2 + b + \epsilon$$

y est la variable expliquée, x_1 et x_2 les variables explicatives. ϵ est l'erreur du modèle, distribué comme une loi normale. a_1, a_2 et b sont les paramètres

à estimer, appelés également *coefficients*. Le modèle peut aussi être écrit sous forme vectorielle : $\mathbf{Y}$ est le vecteur des valeurs de y, $\mathbf{X}$ la matrice dont les colonnes sont les valeurs de x_1 et x_2 et une colonne contenant la valeur 1, $\mathbf{\Theta}$ le vecteur des paramètres et $\mathbf{E}$ le vecteur contenant les résidus :

$$\mathbf{Y} = \mathbf{X\Theta} + \mathbf{E}$$

2 Fabrication des données

On choisit les paramètres du modèle :

```
a1 <- 2
a2 <- 3
b <- 1
Theta <- c(a1, a2, b)
```

On tire 100 valeurs de X entre 0 et 100 de façon uniforme :

```
NbX <- 100
X1 <- runif(NbX) * 100
X2 <- runif(NbX) * 100
X <- cbind(X1, X2, rep(1, length(X1)))
head(X)
```

```
##              X1        X2
## [1,]  4.82511 41.91385 1
## [2,] 21.86699 12.39707 1
## [3,] 50.21846 36.97276 1
## [4,] 40.53148 61.11638 1
## [5,] 39.86500 41.46772 1
## [6,] 94.75279 77.16249 1
```

L'erreur est tirée dans une loi normale centrée d'écart-type 100 :

```
E <- rnorm(nrow(X)) * 100
```

Finalement, on calcule les valeurs de y :

```
Y <- X %*% Theta + E
```

Les données sont représentées sur la figure 1.

3 Estimation du modèle

L'estimation du modèle est faite par la fonction *lm()* de R :

```
Regression <- lm(Y ~ X1 + X2)
summary(Regression)
```

```
##
## Call:
## lm(formula = Y ~ X1 + X2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -243.196  -56.711    1.888   65.774  232.366
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.6636    26.6681   0.250    0.803
## X1            2.0054     0.3724   5.385 5.05e-07
## X2            2.8550     0.3683   7.751 9.04e-12
##
## (Intercept)
## X1          ***
## X2          ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 103.4 on 97 degrees of freedom
## Multiple R-squared:  0.4925, Adjusted R-squared:  0.482
## F-statistic: 47.06 on 2 and 97 DF,  p-value: 5.186e-15
```

Utiliser l'aide de R pour une explication détaillée du fonctionnement de la fonction.

```
help(lm)
```

L'estimation des coefficients se trouve dans le tableau de résultat :

- $\hat{b} = 6.6636$
- $\hat{a}_1 = 2.0054$
- $\hat{a}_2 = 2.855$

Les vraies valeurs sont 1, 2 et 3 : l'estimation n'est pas très bonne parce que l'erreur du modèle a été conçue pour être grande. Les estimateurs ont un intervalle de confiance connu puisque l'erreur du modèle est normale. On note $\sigma_{\hat{a}_1}$ l'écart-type de l'estimateur de a_1 et $t^{\alpha}_{(n-3)}$ la valeur critique de la loi de Student au seuil de risque α à $n-3$ degrés de liberté (n est le nombre d'observations, auquel il faut retirer le nombre de variables explicatives plus 1 pour obtenir le nombre de degrés de liberté du modèle). L'intervalle de confiance est donné par :

$$a_1 = \hat{a}_1 \pm t^{\alpha}_{(n-3)} \frac{\sigma_{\hat{a}_1}}{\sqrt{n}}$$

$t^{\alpha}_{(n-3)}$ vaut à peu près 2 si n n'est pas trop petit, au seuil de risque $\alpha = 5\%$ (dit autrement, au seuil de confiance de 95%). $\sigma_{\hat{a}_1}/\sqrt{n}$ est appelée *erreur standard*

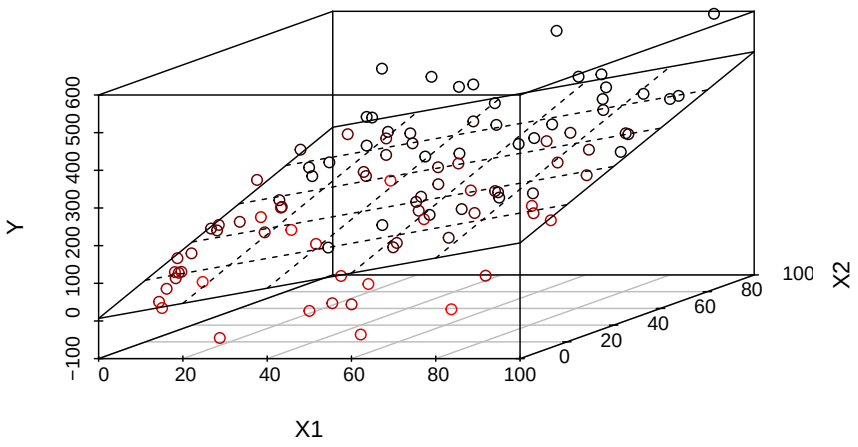

FIGURE 1: Représentation graphique de la régression linéaire à deux variables explicatives. L'estimation du modèle fournit un plan.

de l'estimateur et est affichée dans les sorties de *lm()*. Un test de Student est appliqué à chaque estimateur pour s'assurer qu'il est bien différent de 0 (ce qui signifierait que y ne serait pas lié à cette variable explicative). La probabilité de se tromper en rejetant l'hypothèse de la nullité de l'estimateur est affichée dans la dernière colonne. Classiquement, on retient le coefficient si cette probabilité est inférieure à 5%. Dans notre exemple, la constante est peut-être nulle (on a 80,3% de chance de se tromper en affirmant le contraire), les autres coefficients sont presque certainement non nuls (moins de 0,1% de chances de se tromper, illustré par trois étoiles à côté de la probabilité). Le nuage de points peut être représenté graphiquement avec la librairie *scatterplot3d* :

```
library("scatterplot3d")
s3d <- scatterplot3d(X1, X2, Y, highlight.3d = TRUE)
bhat <- Regression$coefficients[1]
a1hat <- Regression$coefficients[2]
a2hat <- Regression$coefficients[3]
s3d$plane3d(bhat, a1hat, a2hat, lty.box = "solid")
```

La part de la variabilité de Y expliquée par le modèle est quantifiée par son R^2 : 49%. Le reste (51%) est la variabilité non expliquée, celle de l'erreur du modèle : si elle était nulle, tous les points seraient situés sur le plan. La valeur de R^2 est dans les sorties de *lm()*, on peut aussi la calculer directement :

```r
var(X %*% c(a1hat, a2hat, bhat))/var(Y)
```

```
##           [,1]
## [1,] 0.4924724
```

4 Variations

4.1 Erreur du modèle

L'estimation est bien meilleure si l'erreur du modèle est plus faible. Changeons la valeur de **E** :

```r
E <- rnorm(nrow(X)) * 10
```

L'écart-type de l'erreur est maintenant 10 fois plus petit. Le reste du code est inchangé et on obtient :

```r
Y <- X %*% Theta + E
Regression <- lm(formula = Y ~ X1 + X2)
# Figure
s3d <- scatterplot3d(X1, X2, Y, highlight.3d = TRUE)
summary(Regression)
```

```
##
## Call:
## lm(formula = Y ~ X1 + X2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -21.757  -5.809   0.147   5.540  34.462
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.76349    2.53549   1.484    0.141
## X1           1.98483    0.03541  56.059   <2e-16
## X2           2.96553    0.03502  84.680   <2e-16
##
## (Intercept)
## X1          ***
## X2          ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.832 on 97 degrees of freedom
## Multiple R-squared:  0.9912, Adjusted R-squared:  0.991
## F-statistic:  5444 on 2 and 97 DF,  p-value: < 2.2e-16
```

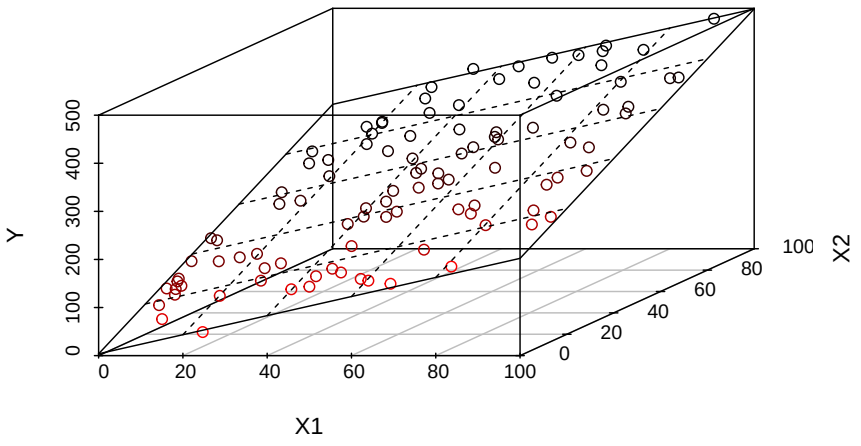

FIGURE 2: Données simulées comme dans la figure précédente, mais avec une erreur 10 fois moins importante. Les points sont très proches du plan de régression.

```
bhat <- Regression$coefficients[1]
a1hat <- Regression$coefficients[2]
a2hat <- Regression$coefficients[3]
s3d$plane3d(bhat, a1hat, a2hat, lty.box = "solid")
```

La variabilité de l'erreur est beaucoup plus faible, donc le R^2 est bien meilleur (plus de 99%). Les paramètres a_1 et a_2 sont très bien estimés. La constante n'est toujours pas estimée correctement : sa valeur réelle est trop proche de 0.

4.2 Nombre de points

La valeur de R^2 peut fortement augmenter en diminuant le nombre de points. Avec deux variables explicatives et trois points, $R^2 = 100\%$ quelles que soient les données (il ne passe qu'un seul plan par trois points). Voici un exemple où 5 points seulement sont utilisés pour estimer le même modèle, avec un écart-type de l'erreur égal à 200 :

```
NbX <- 5
X1 <- runif(NbX) * 100
X2 <- runif(NbX) * 100
X <- cbind(X1, X2, rep(1, length(X1)))
E <- rnorm(nrow(X)) * 200
Y <- X %*% Theta + E
```

```r
Regression <- lm(formula = Y ~ X1 + X2)
summary(Regression)
```

```
##
## Call:
## lm(formula = Y ~ X1 + X2)
##
## Residuals:
##       1       2       3       4       5
## -46.327 -36.915  19.575  56.618   7.049
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  81.7087    96.5662   0.846    0.487
## X1            2.0237     0.8022   2.523    0.128
## X2            2.3040     1.5772   1.461    0.282
##
## Residual standard error: 59.78 on 2 degrees of freedom
## Multiple R-squared:  0.7818, Adjusted R-squared:  0.5637
## F-statistic: 3.584 on 2 and 2 DF,  p-value: 0.2182
```

```r
# Figure
s3d <- scatterplot3d(X1, X2, Y, highlight.3d = TRUE)
bhat <- Regression$coefficients[1]
a1hat <- Regression$coefficients[2]
a2hat <- Regression$coefficients[3]
s3d$plane3d(bhat, a1hat, a2hat, lty.box = "solid")
```

R^2 est proche de 80% mais aucun des coefficients n'est significativement différent de 0. La valeur *ajustée* de R^2 retire de la variance expliquée la part due au simple ajout de variables supplémentaires sans signification : elle permet de comparer la performance d'un modèle à un autre avec un nombre différent de variables. La statistique de Fisher du modèle complet indique une probabilité de se tromper en rejetant l'hypothèse que le modèle n'explique rien supérieure à 20%. R^2 n'est donc pas une indication de la qualité de l'estimation du modèle, seulement de la part de la variance expliquée. Si le nombre d'observations est faible, R^2 peut être grand alors que le modèle n'explique rien. Inversement, l'estimation des paramètres peut être assez bonne avec un R^2 faible si le modèle est est estimé à partir de nombreuses données dans lesquelles la variabilité individuelle est grande. Avec 10000 observations et une erreur d'écart-type 100 :

```r
NbX <- 10000
X1 <- runif(NbX) * 100
X2 <- runif(NbX) * 100
X <- cbind(X1, X2, rep(1, length(X1)))
E <- rnorm(nrow(X)) * 100
Y <- X %*% Theta + E
Regression <- lm(formula = Y ~ X1 + X2)
summary(Regression)
```

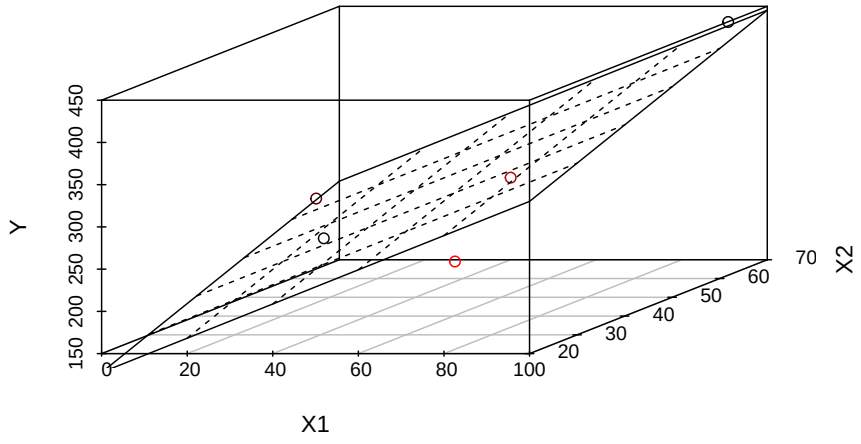

FIGURE 3: Modèle estimé avec 5 points seulement.

```
##
## Call:
## lm(formula = Y ~ X1 + X2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -364.66  -67.41   -0.68   66.85  412.77
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.33230    2.61928   0.509    0.611
## X1           2.00198    0.03468  57.721   <2e-16
## X2           3.02889    0.03463  87.473   <2e-16
##
## (Intercept)
## X1          ***
## X2          ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 99.87 on 9997 degrees of freedom
```

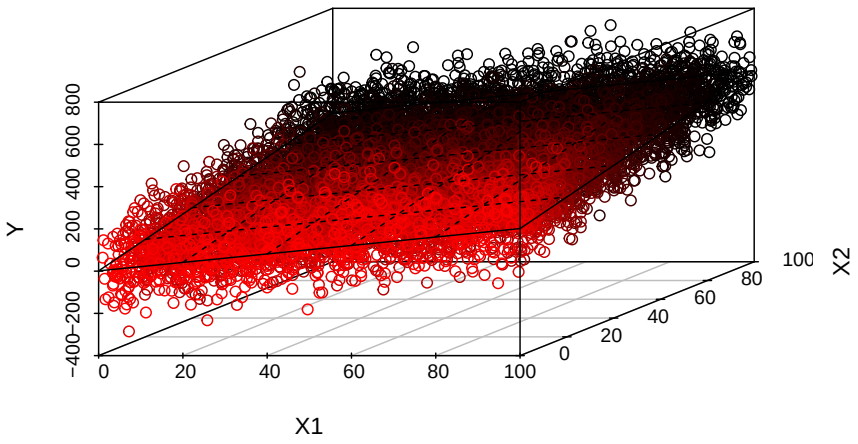

FIGURE 4: Modèle estimé avec 10000 points et une erreur importante.

```
## Multiple R-squared:  0.5286, Adjusted R-squared:  0.5285
## F-statistic:  5606 on 2 and 9997 DF,  p-value: < 2.2e-16
```

```
# Figure
s3d <- scatterplot3d(X1, X2, Y, highlight.3d = TRUE)
bhat <- Regression$coefficients[1]
a1hat <- Regression$coefficients[2]
a2hat <- Regression$coefficients[3]
s3d$plane3d(bhat, a1hat, a2hat, lty.box = "solid")
```

R^2 reste similaire à celui de la première simulation : multiplier les observations n'a pas d'influence. L'estimation des coefficients est 10 fois plus précise qu'avec 100 observations (l'erreur standard est proportionnelle à $1/\sqrt{n}$).

4.3 Calcul des éléments du modèle

Le modèle peut être estimé pas à pas pour prévoir son comportement avant de réaliser l'expérience. A titre d'illustration (figure 5), 100 observations sont simulées, avec une erreur faible (écart-type égal à 10).

```
NbX <- 100
X1 <- runif(NbX) * 100
X2 <- runif(NbX) * 100
X <- cbind(X1, X2, rep(1, length(X1)))
E <- rnorm(nrow(X)) * 10
Y <- X %*% Theta + E
```

```r
Regression <- lm(formula = Y ~ X1 + X2)
summary(Regression)
```

```
##
## Call:
## lm(formula = Y ~ X1 + X2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -24.0098  -6.5104   0.4876   7.4009  18.5557
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.76433    2.93275   0.261    0.795
## X1           1.98997    0.03618  55.000   <2e-16
## X2           3.00778    0.03876  77.609   <2e-16
##
## (Intercept)
## X1          ***
## X2          ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.33 on 97 degrees of freedom
## Multiple R-squared:  0.9892, Adjusted R-squared:  0.989
## F-statistic:  4433 on 2 and 97 DF,  p-value: < 2.2e-16
```

```r
# Figure
s3d <- scatterplot3d(X1, X2, Y, highlight.3d = TRUE)
bhat <- Regression$coefficients[1]
a1hat <- Regression$coefficients[2]
a2hat <- Regression$coefficients[3]
s3d$plane3d(bhat, a1hat, a2hat, lty.box = "solid")
```

La première étape consiste à calculer la matrice $\mathbf{\Sigma} = (\mathbf{X}'\mathbf{X})^{-1}$:

```r
(Sigma <- solve(t(X) %*% X))
```

```
##                X1             X2
## X1  1.227195e-05  2.861349e-07 -0.0006424295
## X2  2.861349e-07  1.408007e-05 -0.0007365698
##    -6.424295e-04 -7.365698e-04  0.0806292031
```

$\mathbf{\Sigma}$ peut être calculée avant de réaliser l'expérience, sans connaître les valeurs de Y. Après l'expérience, les paramètres peuvent être estimés en calculant $\mathbf{\Sigma X}'\mathbf{Y}$:

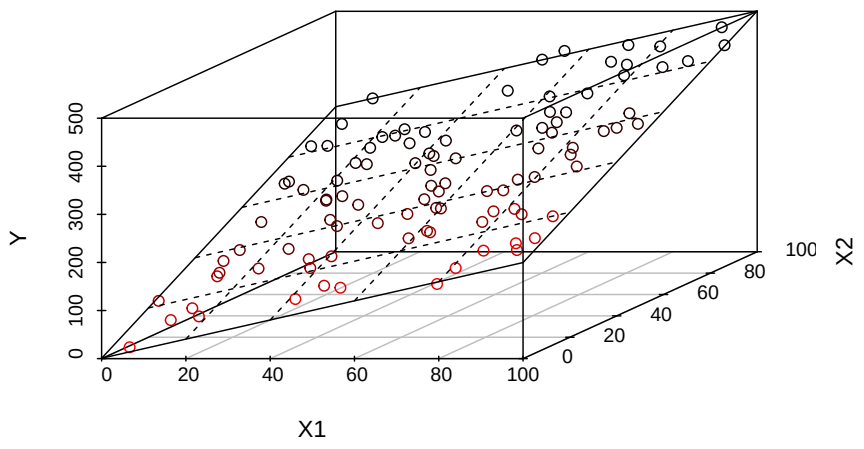

FIGURE 5: Modèle estimé avec 100 points et une faible erreur.

```
Sigma %*% t(X) %*% Y
```

```
##          [,1]
## X1 1.9899651
## X2 3.0077827
##    0.7643349
```

La racine carré de la diagonale de la matrice $\mathbf{\Sigma}$ multipliée par l'écart-type de l'erreur du modèle donne l'erreur standard de l'estimateur des coefficients :

```
(SE <- sqrt(diag(Sigma)) * sd(E))
```

```
##          X1          X2
## 0.03583622 0.03838556 2.90476883
```

L'intervalle de confiance (en plus ou en moins) est obtenu en multipliant l'erreur standard par t :

```
(t <- qt(1 - (1 - 0.95)/2, nrow(X) - ncol(X) - 1))
```

```
## [1] 1.984984
```

```
SE * t
```

```
##          X1          X2
## 0.07113433 0.07619473 5.76592055
```

Les estimateurs des coefficients sont idéalement indépendants entre eux. Ce n'est pas le cas en pratique si les valeurs de $\mathbf{X}$ ne le sont pas. On peut calculer la corrélation entre les estimateurs :

```
round(t(Sigma/sqrt(diag(Sigma)))/sqrt(diag(Sigma)),
    3)
```

```
##         X1     X2
## X1   1.000  0.022 -0.646
## X2   0.022  1.000 -0.691
##     -0.646 -0.691  1.000
```

La corrélation entre $\hat{a}_1$ et $\hat{a}_2$ est égale à 0.022, très proche de 0.
Simulons des valeurs de X très corrélées :

```
X2 <- X1 * (1 + runif(length(X1)))
X <- cbind(X1, X2, rep(1, length(X1)))
# Corrélation entre les coefficients
(Sigma <- solve(t(X) %*% X))
```

```
##                 X1             X2
## X1  8.099497e-05 -4.757553e-05 -5.883953e-04
## X2 -4.757553e-05  3.293277e-05 -2.704209e-05
##    -5.883953e-04 -2.704209e-05  4.211929e-02
```

```
round(t(Sigma/sqrt(diag(Sigma)))/sqrt(diag(Sigma)),
    3)
```

```
##         X1     X2
## X1   1.000 -0.921 -0.319
## X2 -0.921  1.000 -0.023
##    -0.319 -0.023  1.000
```

L'estimation du modèle reste bonne :

```
Y <- X %*% Theta + E
Regression <- lm(formula = Y ~ X1 + X2)
summary(Regression)
```

```
##
## Call:
## lm(formula = Y ~ X1 + X2)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -23.8436  -6.3916   0.4818   7.8284  18.6514
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.22759    2.10554   0.583    0.561
## X1           2.08854    0.09233  22.620   <2e-16
## X2           2.93166    0.05888  49.794   <2e-16
##
## (Intercept)
## X1          ***
## X2          ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.26 on 97 degrees of freedom
## Multiple R-squared:  0.9971, Adjusted R-squared:  0.997
## F-statistic: 1.673e+04 on 2 and 97 DF,  p-value: < 2.2e-16
```

Les estimateurs de $\hat{a}_1$ et $\hat{a}_2$ ont pourtant une corrélation égale à -0.921 : toute augmentation dans l'estimateur de l'un entraîne une diminution presque identique de l'estimateur de l'autre. La régression linéaire est très robuste face à la violation de ses hypothèses.

5 Design expérimental

5.1 Performance

Les valeurs de $\mathbf{Y}$, et donc l'estimation des paramètres du modèle et de son erreur, ne sont connues qu'après l'expérience. Mais les valeurs de $\mathbf{X}$ sont connues avant : leur choix est le design expérimental. L'objectif principal de l'estimation d'un modèle linéaire est l'estimation aussi précise que possible de ses paramètres. Pour cela, l'erreur standard doit être aussi faible et le nombre d'observation aussi grand que possible. L'erreur standard des estimateurs est donnée par la diagonale de $\mathbf{\Sigma} = (\mathbf{X}'\mathbf{X})^{-1}$ multipliée par l'écart-type de l'erreur du modèle. L'erreur du modèle dépendra des données, mais le design permet de minimiser la diagonale de $\mathbf{\Sigma}$. L'erreur standard est minimale pour tous les coefficients si toutes les combinaisons des valeurs extrêmes des variables explicatives sont utilisées (Cochran & Cox, 1992) : ce design est appelé *design factoriel*. Le nombre de combinaison est 2 à la puissance le nombre de facteurs ; dans notre exemple,

les valeurs seraient :

$$\mathbf{X} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 100 & 1 \\ 100 & 100 & 1 \\ 100 & 100 & 1 \end{pmatrix}$$

La valeur de l'erreur standard de chaque variable est connue dans ce cas : elle est égale à la moitié de l'écart entre les deux valeurs extrêmes de la variable explicative multipliée par le nombre d'observations.

Si le design n'est pas factoriel, l'erreur standard des coefficients sera plus grande. Le rapport entre l'erreur standard du design factoriel est celle du design choisi est appelé la performance du design (Baraloto *et al.*, 2010). Avec les 100 observations uniformément distribuées de la simulation précédente, le calcul est le suivant :

```
# Facteur d'échelle = écart-type des deux extrêmes
S <- apply(X, 2, function(Xj) (max(Xj) - min(Xj))/2)
# Performance
(P <- 1/(S * sqrt(diag(Sigma) * nrow(X))))
```

```
##         X1         X2
## 0.2275832 0.2005624        Inf
```

Dans cet exemple, les valeurs des variables explicatives ont été choisies uniformément entre 0 et 100. La performance du design est de 23% pour x_1 et 20% pour x_2, ce qui signifie que l'erreur standard de l'estimateur de a_2 est 5 fois plus grande que dans le design factoriel. Toutes choses égales par ailleurs, il faudra 25 fois plus d'observations pour obtenir le même intervalle de confiance pour cet estimateur. La dernière valeur (*Inf*) n'a pas de signification.

La performance est nettement améliorée en se rapprochant du design factoriel :

```
# Tirage de 2500 valeurs
NbX <- 2500
X1 <- runif(NbX) * 100
X2 <- runif(NbX) * 100
X <- cbind(X1, X2, rep(1, length(X1)))
E <- rnorm(nrow(X)) * 100
Y <- X %*% Theta + E
# Elimination des valeurs intermédiaires (4% des
# points sont conservés)
Extreme <- (X1 < 10 | X1 > 90) & (X2 < 10 | X2 > 90)
X1 <- X1[Extreme]
X2 <- X2[Extreme]
X <- cbind(X1, X2, rep(1, length(X1)))
E <- E[Extreme]
Y <- X %*% Theta + E
(Sigma <- solve(t(X) %*% X))
```

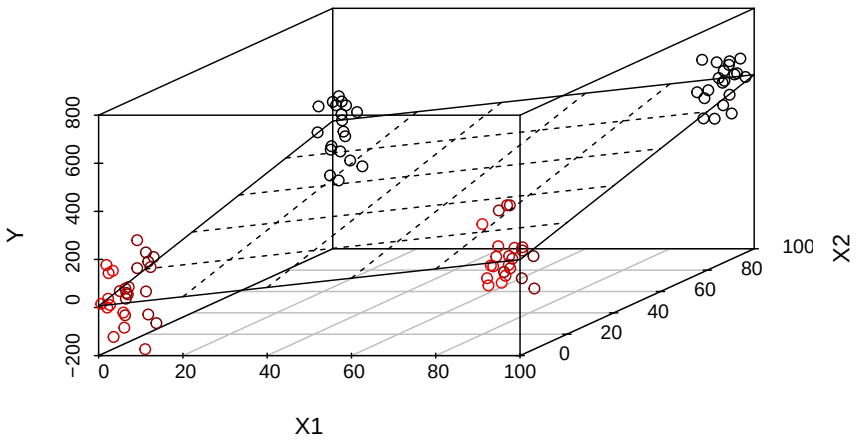

FIGURE 6: Elimination des valeurs intermédiaires des facteurs.

```
##                X1             X2
## X1  5.479472e-06 -2.920513e-07 -0.0002496656
## X2 -2.920513e-07  5.638243e-06 -0.0002361176
##    -2.496656e-04 -2.361176e-04  0.0336762622
```

```
# Facteur d'échelle = écart-type des deux extrêmes
S <- apply(X, 2, function(Xj) (max(Xj) - min(Xj))/2)
# Performance
(P <- 1/(S * sqrt(diag(Sigma) * nrow(X))))
```

```
##         X1         X2
## 0.9135162 0.8954324       Inf
```

```
# Figure
Regression <- lm(formula = Y ~ X1 + X2)
s3d <- scatterplot3d(X1, X2, Y, highlight.3d = TRUE)
bhat <- Regression$coefficients[1]
a1hat <- Regression$coefficients[2]
a2hat <- Regression$coefficients[3]
s3d$plane3d(bhat, a1hat, a2hat, lty.box = "solid")
```

La performance est ici proche 90% pour les deux variables.

6 Conclusion

Choisir un design proche du design factoriel permet de minimiser l'erreur standard de l'estimation des paramètres du modèle. La contrepartie est la perte d'information sur les valeurs intermédiaires : aucune information sur la linéarité du modèle n'est disponible si les seules valeurs extrêmes des facteurs sont retenues. Ajouter des valeurs intermédiaires règle ce problème, mais a un coût en terme de performance qui peut être calculé.

La corrélation entre les estimateurs est un problème plus théorique que pratique. Son effet n'est pas énorme sur l'estimation, mais elle doit être évitée autant que possible. Elle peut être calculée dès la construction de l'expérimentation.

Le nombre d'observations est un choix économique : chaque fois qu'il est quadruplé, la précision est doublée (l'erreur standard est divisée par 2). Il est donc très rentable de travailler sur le design expérimental.

Références

Baraloto, C., Marcon, E., Morneau, F., Pavoine, S. & Roggy, J.C. (2010) Integrating functional diversity into tropical forest plantation designs to study ecosystem processes. *Annals of Forest Science*, **67**, 303.

Cochran, W.G. & Cox, G.M. (1992) *Experimental Designs.* John Wiley & Sons, New York, 2nd ed. edition.