

Régression de modèle II

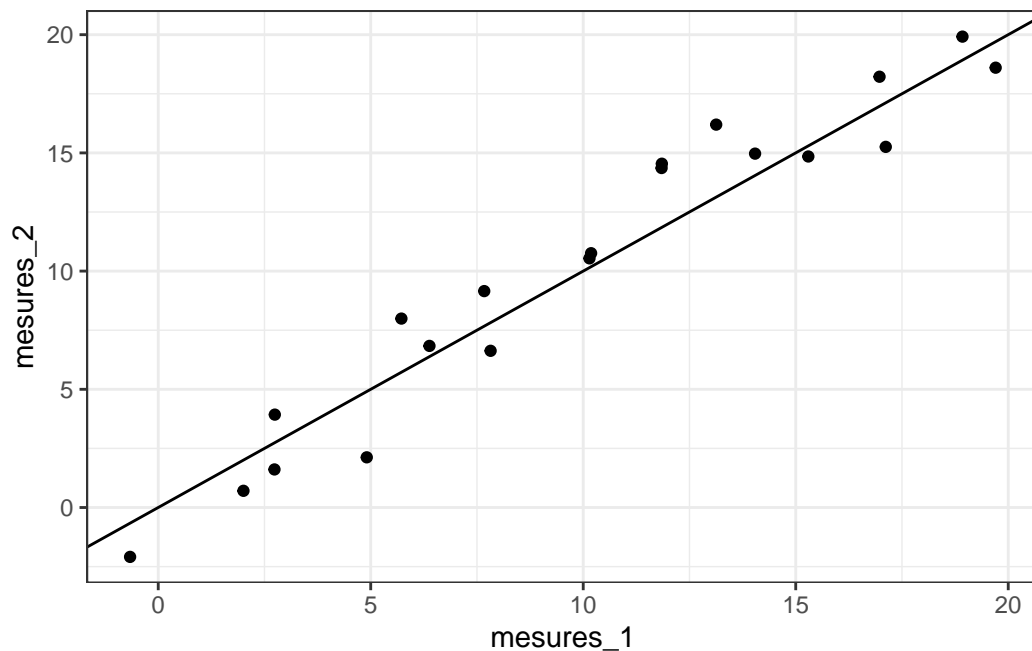
Théorie

La régression de modèle II (package **lmodel2**) s'applique aux données pour lesquelles X et Y sont entachés d'erreurs. L'application typique est la comparaison de mesures effectuées par deux instruments différents.

Exemple

Soient 20 mesures dont les vraies valeurs sont 1:20. Deux instruments sont utilisés, chacun comporte une erreur normale. On compare les deux jeux de mesures pour valider la possibilité d'utiliser indifféremment l'un ou l'autre des instruments. Les seules données disponibles sont les mesures : les valeurs exactes ne sont pas connues des expérimentateurs.

```
library("tidyverse")
mesures_n <- 20
mesures_exactes <- 1:mesures_n
erreur_1 <- 1
erreur_2 <- 2
# Création des données
mesures <- tibble(
  mesures_1 = mesures_exactes + rnorm(mesures_n, sd = erreur_1),
  mesures_2 = mesures_exactes + rnorm(mesures_n, sd = erreur_2)
)
# Figure
mesures %>%
  ggplot(aes(x = mesures_1, y = mesures_2)) +
  geom_point() +
  geom_abline(slope = 1, intercept = 0)
```



Régression

La régression utilise la fonction `lmodel2::lmodel2()`.

```
library("lmodel2")
mesures %>%
  lmodel2(mesures_2 ~ mesures_1, data = .) -> mesures_lm2
# Résultats
mesures_lm2
```

Model II regression

Call: `lmodel2(formula = mesures_2 ~ mesures_1, data = .)`

n = 20 r = 0.9689866 r-square = 0.9389351
 Parametric P-values: 2-tailed = 2.252573e-12 1-tailed = 1.126286e-12
 Angle between the two OLS regression lines = 1.797324 degrees

Regression results

| | Method | Intercept | Slope | Angle (degrees) |
|---|--------|------------|----------|-----------------|
| 1 | OLS | -0.2808596 | 1.061350 | 46.70475 |
| 2 | MA | -0.6497018 | 1.098507 | 47.68758 |
| 3 | SMA | -0.6180635 | 1.095320 | 47.60469 |

P-perm (1-tailed)

| | |
|---|----|
| 1 | NA |
| 2 | NA |

3

NA

Confidence intervals

| | Method | 2.5%-Intercept | 97.5%-Intercept |
|---|--------|----------------|-----------------|
| 1 | OLS | -1.825250 | 1.2635305 |
| 2 | MA | -2.132077 | 0.6408058 |
| 3 | SMA | -2.029661 | 0.6313287 |

| | 2.5%-Slope | 97.5%-Slope |
|---|------------|-------------|
| 1 | 0.9273178 | 1.195383 |
| 2 | 0.9685028 | 1.247840 |
| 3 | 0.9694575 | 1.237523 |

Eigenvalues: 78.03967 1.218742

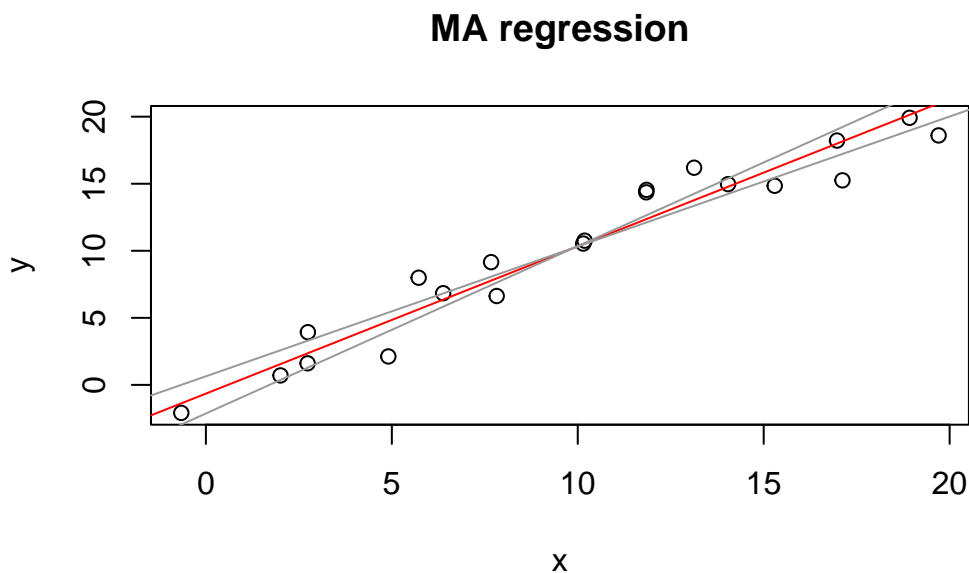
H statistic used for computing C.I. of MA: 0.003951986

La méthode à utiliser ici est celle de l'axe majeur (MA : voir la vignette du package (section I) pour le choix de la bonne méthode).

L'ordonnée à l'origine n'est pas différente de 0 et la pente n'est pas différente de 1 : les deux instruments fournissent la même mesure.

Une figure peut être produite :

```
plot(mesures_lm2)
```



Avec ggplot :

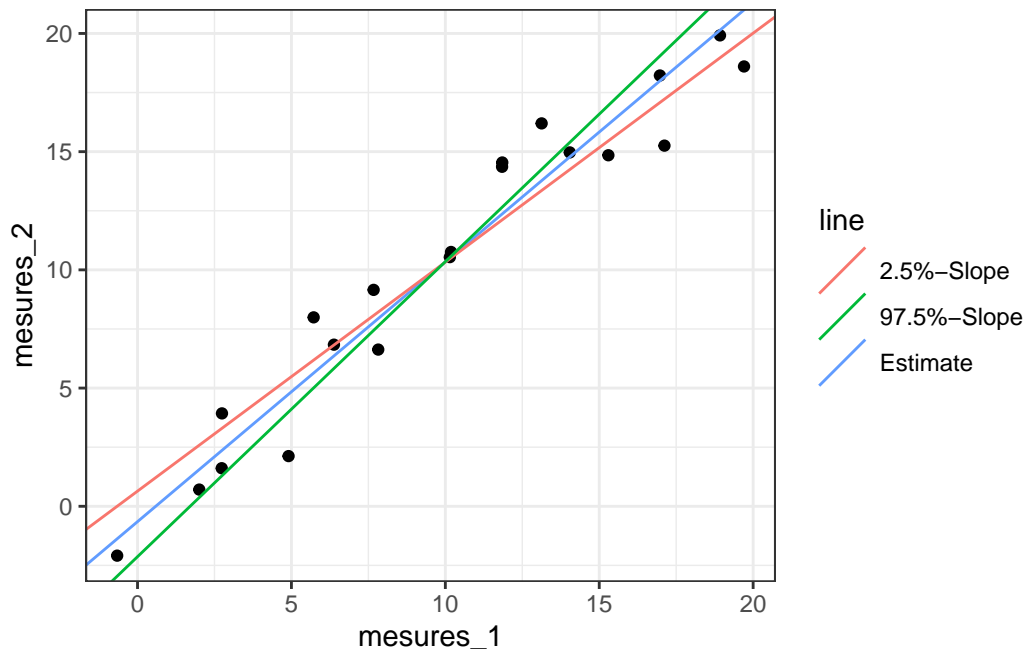
```

lmodel2_lines <- function(model, method = "MA") {
  # Code adapté de lmodel2::plot.lmodel2()
  if ((method != "OLS") && (model$rsquare <= model$epsilon)) {
    # Pas de corrélation entre x et y
    stop("R-square = 0: model and C.I. not drawn for MA, SMA or RMA")
  }
  # Centre de gravité du nuage de points
  centroid <- colMeans(as.data.frame(model[c("x", "y")]))
  # Modèle utilisé
  row <- which(model$regression.results == method)
  # Pente estimée et intervalle de confiance
  slope <- c(
    model$regression.results[row, "Slope"],
    model$confidence.intervals[row, "2.5%-Slope"],
    model$confidence.intervals[row, "97.5%-Slope"]
  )
  # Intercept correspondant (pour l'IC)
  intercept <- centroid[2] - slope * centroid[1]
  # Intercept estimé
  intercept[1] <- model$regression.results[row, "Intercept"]

  return(
    tibble(line = c("Estimate", "2.5%-Slope", "97.5%-Slope"), slope, intercept)
  )
}

ggplot() +
  geom_point(data = mesures, mapping = aes(x = mesures_1, y = mesures_2)) +
  geom_abline(
    data = lmodel2_lines(mesures_lm2),
    mapping = aes(slope = slope, intercept = intercept, color = line)
  )

```



La proportion de la variance estimée est :

```
mesures_lm2$rsquare
```

```
[1] 0.9389351
```

Régression classique

Le modèle classique sous-estime la pente de la relation (pente en gras et intervalle de confiance grisé).

```
mesures %>%  
  ggplot(aes(x = mesures_1, y = mesures_2)) +  
  geom_point() +  
  geom_smooth(method = "lm") +  
  geom_abline(  
    data = lmodel2_lines(mesures_lm2),  
    mapping = aes(slope = slope, intercept = intercept, color = line)  
  )
```

