

Modèles Linéaires

Correction Séance 3 Licence 3 MIASHS (2021-2022)

Guillaume Metzler

Université de Lyon, Université Lumière Lyon 2
Laboratoire ERIC UR 3083, Lyon, France

guillaume.metzler@univ-lyon2.fr

Dans cette séance, nous allons traiter les exercices 3 des feuilles de TD 1 et 2.

L'objectif sera d'étudier quelques propriétés du modèles linéaire simple et du modèle linéaire multiple en prenant un cas particulier, un modèle linéaire multiple *quadratique*.

1 Quelques rappels

Notre modèle linéaire multiple peut s'écrire sous la forme

$$\forall i \in \llbracket 1, n \rrbracket, y_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \cdots + \beta_p x_{i,p} + \varepsilon_i,$$

où

- y est la variable réponse (ou la variable à prédire)
- x est la variable prédictive
- β_0 et β_1 sont les paramètres du modèles
- ε est un bruit blanc gaussien qui représente l'erreur de modélisation.

Ce modèle peut facilement se réécrire sous la forme suivante

$$Y = X\beta + \varepsilon,$$

avec les notations vectorielles/matricielles suivantes

$$Y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \in \mathbb{R}^n, \beta = \begin{pmatrix} \beta_0 \\ \vdots \\ \beta_p \end{pmatrix} \in \mathbb{R}^{p+1}, X = \begin{pmatrix} 1 & x_{1,1} & \cdots & x_{1,p} \\ \vdots & \vdots & & \vdots \\ 1 & x_{n,1} & \cdots & x_{n,p} \end{pmatrix} \in \mathcal{M}_{n,p+1}(\mathbb{R}), \varepsilon = \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{pmatrix} \in \mathbb{R}^n.$$

On rappelle également l'expression des solutions

Proposition 1.1: Solution du modèle linéaire (multiple)

Placons-nous sous les hypothèses du modèle linéaire gaussien vues lors de la précédente séance et considérons le modèle :

$$Y = \beta X + \varepsilon.$$

Si on dispose d'un n -échantillon $(y_i, x_i)_{i=1}^n$ d'individus indépendants alors la solution du problème des *moindres carrés ordinaires*, *i.e.* du problème

$$\min_{\beta \in \mathbb{R}^{p+1}} \|Y - \hat{Y}\|_2^2.$$

est donnée par

$$\hat{\beta} = (X^T X)^{-1} X^T Y.$$

2 Retour au TD

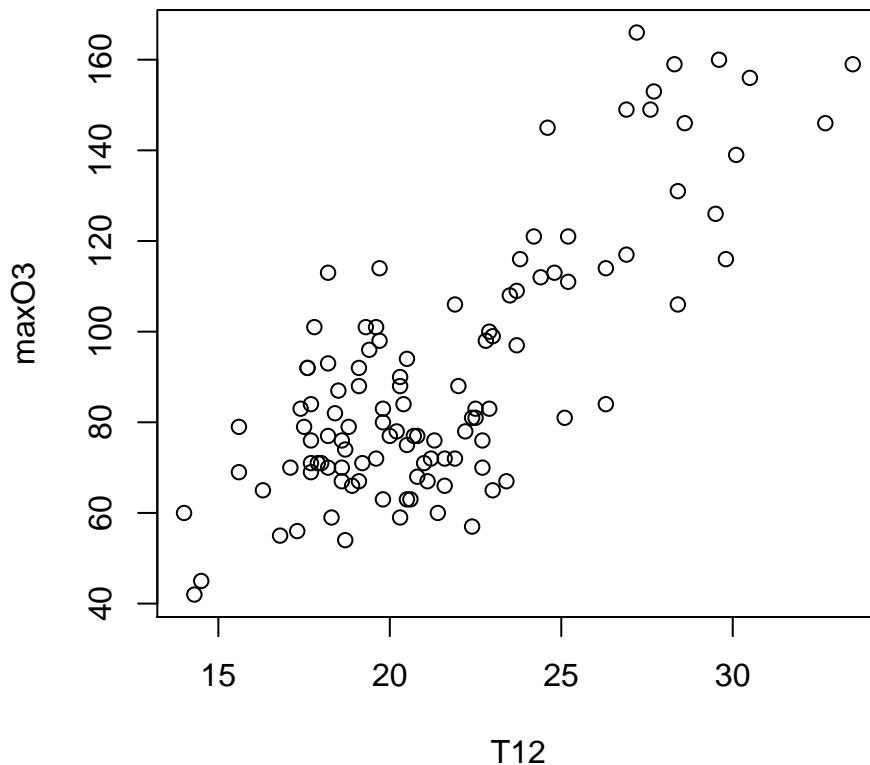
On va travailler avec le jeu de données *ozone* que vous pourrez trouver [ici](#) et que vous pourrez importer de la façon suivante

```
ozone <- read.table("data/ozone.csv",
                    quote="\"",
                    comment.char=" ",
                    header = TRUE)
```

Dans un premier temps, on se concentre sur le modèle linéaire simple dans lequel on va chercher à prédire la valeur de la variable *maxO3*, concentration maximale en O_3 , en fonction de la température à midi *T12*.

Regardons tout d'abord notre nuage de points

```
plot(maxO3~T12, data=ozone)
```



Les données semblent globalement être ajustées sur une droite, *i.e.* on dégage une tendance globalement linéaire dans nos données, ce qui justifie l'usage d'un modèle linéaire simple.

On va ensuite estimer les paramètres du modèle

```
mymodel <- lm(maxO3~T12, data = ozone)
resume <- summary(mymodel)
```

On peut commencer par regarder les coefficients de la régression, qui, on le rappelle sont donnés par

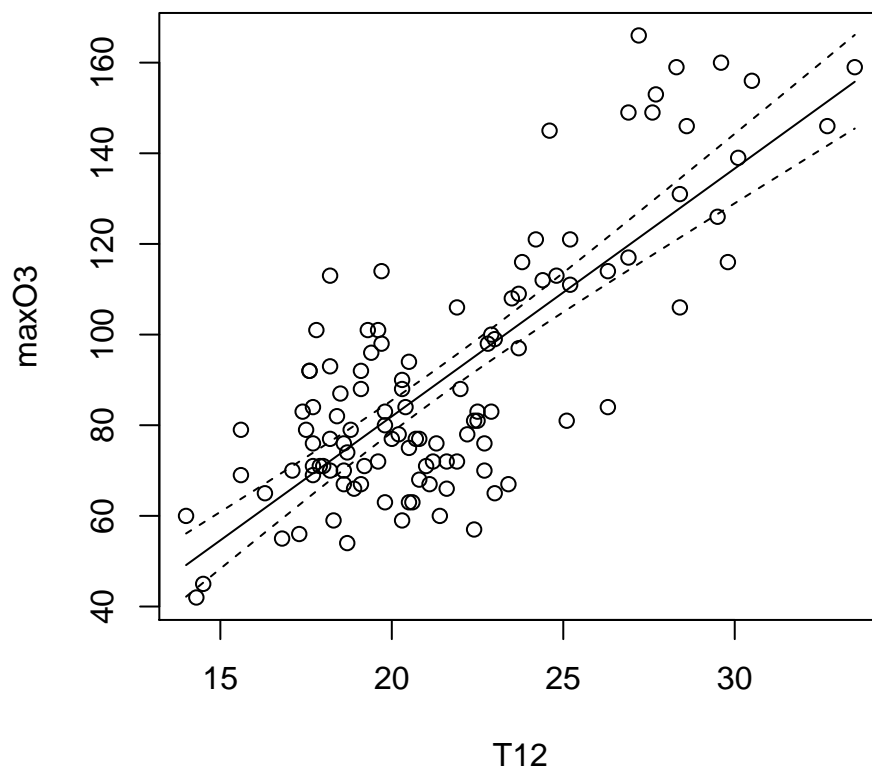
$$\hat{\beta}_1 = \frac{Cov[X, Y]}{Var[X]} \quad \text{et} \quad \hat{\beta}_0 = \mathbb{E}[Y] - \hat{\beta}_1 \mathbb{E}[X].$$

```
mymodel$coefficients
```

```
## (Intercept)      T12  
## -27.419636    5.468685
```

On cherche ensuite à tracer l'estimation de la droite de régression, ainsi qu'un intervalle de confiance à 95% de celle-ci.

```
plot(maxO3~T12,data=ozone)  
T12=seq(min(ozone[, "T12"]),max(ozone[, "T12"]),length=100)  
grille<-data.frame(T12)  
ICdte<-predict(mymodel,new=grille,interval="confidence",level=0.95)  
matlines(grille$T12,cbind(ICdte),lty=c(1,2,2),col=1)
```



Les droites en pointillés définissent l'intervalle de confiance sur la droite de régression. Il correspond donc à l'ensemble des modèles, *i.e.* l'ensemble des droites que nous

obtiendrons, dans 95% des cas, à partir d'un échantillon issu de la même distribution que nos données actuelles.

La droite capte la tendance globale présente dans les données, en revanche ce modèle simple est dans l'incapacité de capter la variabilité présente aux extrémités du graphe.

On cherche ensuite à représenter le vecteur des résidus. On rappelle que ces derniers sont définis par

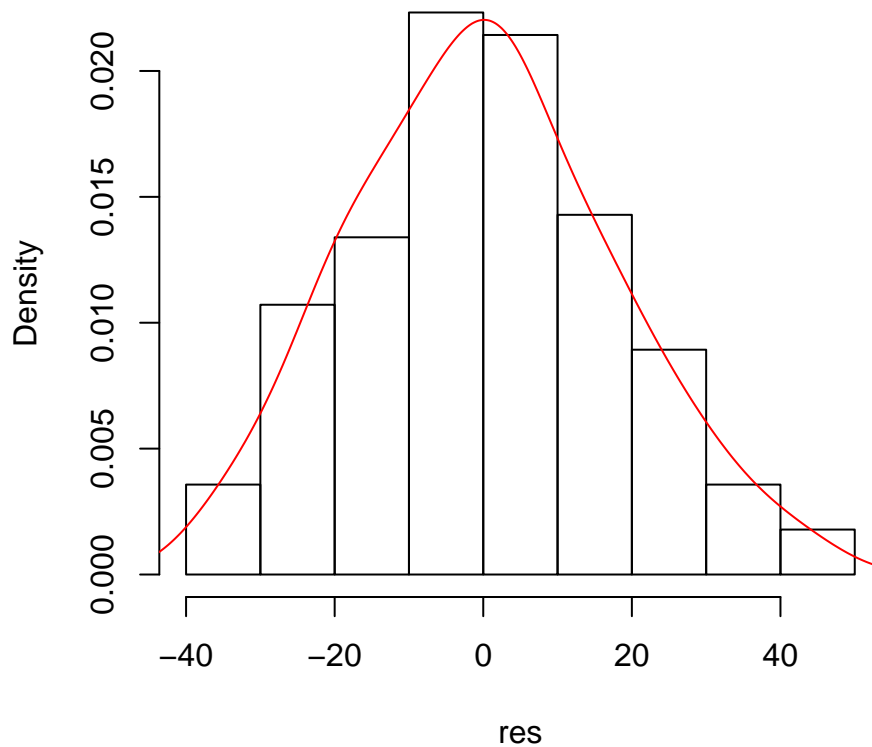
$$e_i = y_i - \hat{y}_i = y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i).$$

```
# Récupération des résidus  
res <- mymodel$residuals
```

On va ensuite les représenter graphiquement de plusieurs façons, ici on va choisir de le faire sous la forme d'un histogramme et on va vérifier qu'ils sont distribués selon une loi gaussienne centrée.

```
# Représentation de la distribution des résidus  
hist(res, probability = TRUE)  
lines(density(res), col = "red")
```

Histogram of res



On va maintenant explorer les sorties de la fonction `summary` de notre modèle linéaire.

```
summary(mymodel)

##
## Call:
## lm(formula = maxO3 ~ T12, data = ozone)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -38.079 -12.735   0.257  11.003  44.671
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -27.4196     9.0335  -3.035   0.003 **
```

```
## T12          5.4687      0.4125  13.258   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17.57 on 110 degrees of freedom
## Multiple R-squared:  0.6151, Adjusted R-squared:  0.6116
## F-statistic: 175.8 on 1 and 110 DF,  p-value: < 2.2e-16
```

Regardons les différentes sorties de cette fonction :

- Les premières informations décrivent le modèle appris, la distribution des résidus en indiquant les informations : *min-max* ainsi que les différents quartiles de la distribution des résidus.
- On retrouve ensuite des informations relatives aux paramètres de la droite de régression :
Estimate : qui donne les valeurs estimées à l'aide de notre jeu de données pour les différents paramètres.
Erreur standard : il s'agit de l'écart type de notre estimateur. Par exemple, dans le cas du modèle linéaire simple, ces erreurs standards sont données par

$$\sqrt{\frac{\hat{\sigma}^2 \sum_{i=1}^n x_i^2}{n \sum_{i=1}^n (x_i - \bar{x})^2}} \quad \text{et} \quad \sqrt{\frac{\hat{\sigma}^2}{\sum_{i=1}^n (x_i - \bar{x})^2}}.$$

pour le paramètre β_0 et β_1 respectivement.

Ce que l'on peut calculer comme suit

```
# Calcul des erreurs standards

n = length(res)
sigma_hat = sum(res^2)/(n-2)
s2_x = var(ozone$T12)*(n-1)
sum2_x = sum(ozone$T12^2)

# Pour le paramètre beta_0


err_beta_0_hat <- sqrt(sigma_hat*sum2_x/(s2_x*n))
err_beta_0_hat
## [1] 9.033494
# Pour le paramètre beta_1

err_beta_1_hat <- sqrt(sigma_hat/s2_x)
err_beta_1_hat
```

```
## [1] 0.4124939
```

t.value : il s'agit des valeurs des statistiques de tests utilisées lorsque l'on cherche à déterminer si les paramètres de la régression sont significatifs. Pour β_0 et β_1 ces valeurs sont données par

$$\frac{\hat{\beta}_0}{\sqrt{\frac{\hat{\sigma}^2 \sum_{i=1}^n x_i^2}{n \sum_{i=1}^n (x_i - \bar{x})^2}}} \quad \text{et} \quad \frac{\hat{\beta}_1}{\sqrt{\frac{\hat{\sigma}^2}{\sum_{i=1}^n (x_i - \bar{x})^2}}}.$$

On peut calculer sur  comme suit :

```
# Pour le paramètre beta_0
```

```
beta_0_hat <- mymodel$coefficients[1]  
var_beta_0_hat <- sigma_hat*sum2_x/(s2_x*n)
```

```
t_beta_0 = beta_0_hat/sqrt(var_beta_0_hat)  
t_beta_0  
## (Intercept)  
## -3.03533
```

```
# Pour le paramètre beta_1
```

```
beta_1_hat <- mymodel$coefficients[2]  
var_beta_1_hat <- sigma_hat/s2_x  
  
t_beta_1 = beta_1_hat/sqrt(var_beta_1_hat)  
t_beta_1  
## T12  
## 13.25761
```

p.value : on y trouve les p-value associées au test de significativité des paramètres, elle sont données par

```
# Pour le paramètre beta_0
```

```
2*(1-pt(abs(t_beta_0),n-2))  
## (Intercept)  
## 0.002999431
```

```
# Pour le paramètres beta_1
```

```
2*(1-pt(abs(t_beta_1),n-2))  
## T12  
## 0
```

- Enfin, on dispose d'informations sur les résidus, ce que l'on appelle le *residual*

standard error qui est défini par

$$\sqrt{\frac{1}{n-2} \sum_{i=1}^n (y_i - \hat{y}_i)^2},$$

avec un nombre de degrés de liberté qui est égal à $n - p$ ($p = 2$ ici).

On donne aussi le *coefficient de détermination* du modèle, noté R^2 que l'on définit par

$$R^2 = \frac{SSE}{SST} = \frac{SST - SSR}{SST},$$

où

$SSE = \sum_{i=1}^n (y_i - \bar{y})^2$ est appelée *Sum of Squared Explained*. Il s'agit de la variation expliquée par le modèle.

$SSR = \sum_{i=1}^n (y_i - \hat{y}_i)^2$ est appelée *Sum of Squared Résiduals*. Il s'agit de la variation non expliquée par le modèle. C'est une variation due aux erreurs du modèle.

$SST = \sum_{i=1}^n (y_i - \bar{y})^2$ est appelée *Sum of Squares Total*. Il s'agit de la variation totale de notre jeu de données.

On a la relation

$$SST = SSE + SSR.$$

Le R^2 ajusté est donné par

$$R_{adjusted}^2 = 1 - \frac{(n-1)(1-R^2)}{n-p-1}.$$

Finalement un test de Fisher est réalisé pour tester la significativité globale du modèle, *i.e.* on réalise le test suivant :

$$H_0 : \beta_j = 0 \forall j \quad \text{versus} \quad H_1 = \exists j \text{ s.t. } \beta_j \neq 0.$$

La statistique de test employée est alors donnée par

$$\frac{SSE}{SSR} = \frac{(SST - SSR)/p}{(SSR)/(n-p-1)}.$$

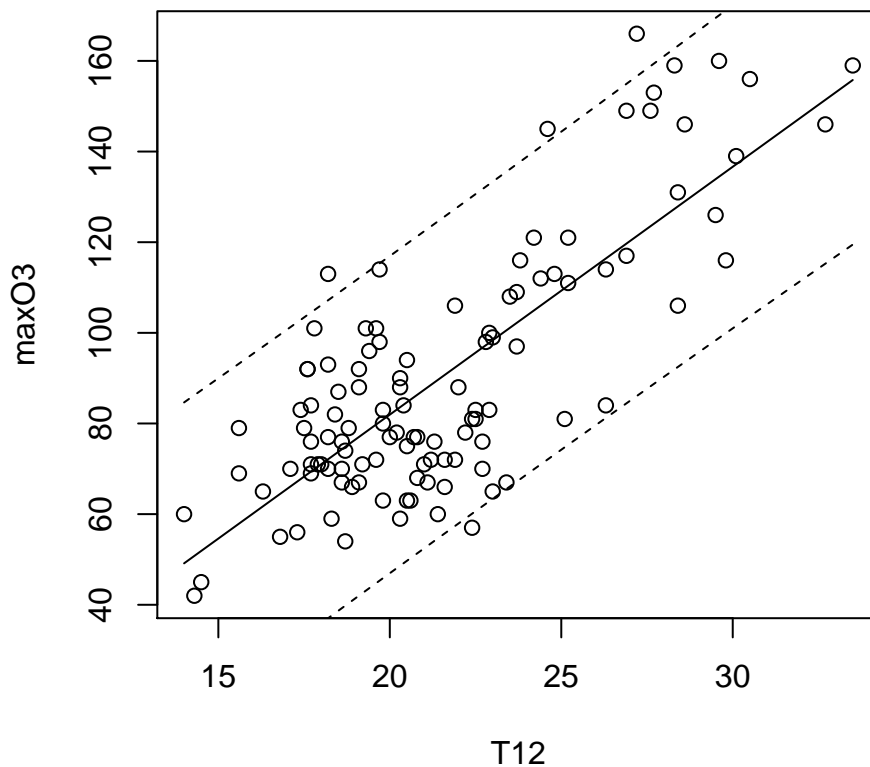
Cette statistique de test suit une loi de Fisher à $p, n - p - 1$ degrés de liberté.

On s'intéresse à la qualité de la prévision du modèle, on va donc représenter l'intervalle de prédiction du modèle

```

plot(maxO3~T12,data=ozone)
T12=seq(min(ozone[, "T12"]),max(ozone[, "T12"]),length=100)
grille<-data.frame(T12)
ICprev<-predict(mymodel,new=grille,interval="pred",level=0.95)
matlines(grille$T12,cbind(ICprev),lty=c(1,2,2),col=1)

```



En fonction du modèle appris, on observe qu'environ 95% des données doivent se trouver dans cet intervalle de prédiction.

On termine en déterminant les intervalles de confiance des paramètres de la régression

$$IC_{1-\alpha} = \left[\hat{\beta} - t_{1-\alpha/2, n-2} \sqrt{\text{Var}[\hat{\beta}]}, \hat{\beta} + t_{1-\alpha/2, n-2} \sqrt{\text{Var}[\hat{\beta}]} \right].$$

On va calculer les différentes quantités pour les estimateurs

```

# Pour le paramètre beta_1

beta_1_hat <- mymodel$coefficients[2]
var_beta_1_hat <- sigma_hat/s2_x

# Borne Inf
beta_1_hat - qt(0.975,n-2)*sqrt(var_beta_1_hat)

##      T12
## 4.651219

# Borne Sup
beta_1_hat + qt(0.975,n-2)*sqrt(var_beta_1_hat)

##      T12
## 6.286151

# Pour le paramètre beta_0

beta_0_hat <- mymodel$coefficients[1]
var_beta_0_hat <- sigma_hat*sum2_x/(s2_x*n)

# Borne Inf
beta_0_hat - qt(0.975,n-2)*sqrt(var_beta_0_hat)

## (Intercept)
## -45.3219

# Borne Sup
beta_0_hat + qt(0.975,n-2)*sqrt(var_beta_0_hat)

## (Intercept)
## -9.517371

```

On va maintenant s'intéresser à un modèle linéaire simple avec deux variables : $T12$ ainsi que le carré de cette valeur, *i.e.* en utilisant un modèle *quadratique*. Ce modèle sera de la forme

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \varepsilon_i.$$

On peut à nouveau écrire ce modèle sous forme matricielle comme suit :

$$Y = X\beta + \varepsilon,$$

où

$$Y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \in \mathbb{R}^n, \beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{pmatrix} \in \mathbb{R}^3, X = \begin{pmatrix} 1 & x_1 & x_1^2 \\ \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 \end{pmatrix} \in \mathcal{M}_{n,3}(\mathbb{R}), \varepsilon = \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{pmatrix} \in \mathbb{R}^n.$$

On va construire notre modèle et estimer les paramètres de la régression.

```
# On va créer notre variable température au carré
ozone$T12_square <- ozone$T12^2
# puis estimer les paramètres de notre régression
mymodel2 <- lm(maxO3~T12 + T12_square, data = ozone)
# Analyse de la régression
summary(mymodel2)

##
## Call:
## lm(formula = maxO3 ~ T12 + T12_square, data = ozone)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -34.401 -13.490  -1.266   11.555   45.061
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  76.74824    41.75410   1.838  0.0688 .
## T12          -3.87486     3.68286  -1.052  0.2951
## T12_square    0.20219     0.07922   2.552  0.0121 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17.14 on 109 degrees of freedom
## Multiple R-squared:  0.6368, Adjusted R-squared:  0.6301
## F-statistic: 95.54 on 2 and 109 DF, p-value: < 2.2e-16
```

On se propose ensuite de comparer le modèle linéaire simple avec ce modèle linéaire quadratique à l'aide du *BIC*. Le BIC d'un modèle est défini

$$BIC = 2\log(2\pi\hat{\sigma}^2) + p\log(n), \quad (1)$$

où p désigne le nombre de paramètres que l'on a à apprendre.

Pour se convaincre de cette relation qui part de la définition suivante du BIC :

$$BIC = 2\ln(\hat{L}) + p\ln(n)$$

où \hat{L} est la valeur du maximum de vraisemblance. A titre d'exercice et d'entraînement, il est fortement conseillé de reprendre les calculs effectués à la Section 3.5 de votre cours afin de retrouver l'Equation (1).