# STA201: TD5 - Régression linéaire multiple

*christine.keribin@math.u-psud.fr, Baptiste Broto, Timothée Mathieu, S. Thépaut*

*10 octobre 2018*

## *Eléments de corrigé*

Objectifs: de la théorie à la pratique

- Résoudre un cas de régression linéaire multiple sur un jeu de données réelles
- Programmer dans R avec des matrices
- Comprendre l'utilisation de la fonction `lm`, savoir l'utiliser et interpréter ses sorties en correspondance avec les résultats théoriques

Vous trouverez sur le site pédagogique du cours le fichier `STA201-TP5-reglin-init.R` qui contient les commandes inscrites dans l'énoncé et le fichier `hFE.csv` des données traitées dans ce TP. Commencer par mettre en place l'environnement de travail:

```
setwd("Mon répertoire") # à personnaliser
rm(list=objects())
graphics.off()
```

# 1 Utiliser les matrices dans R

La régression fait largement appel à l'algèbre linéaire. R permet la définition de matrices, objets à deux dimensions, dont les composantes sont de mode homogène (ne contiennent que des éléments de même nature). La création d'une matrice se fait avec la syntaxe :

```
n=4 # nombre de lignes
p=3 # nombre de colonnes
matrix(vec,nrow=n,ncol=p)
```

où `vec` est le vecteur contenant les éléments de la matrice, qui seront rangés en colonne (sauf si l'argument `byrow=TRUE` est choisi). On peut également créer des matrices en concaténant des vecteurs de même dimension en ligne (`rbind()`) ou en colonne (`cbind()`).

1. Créer les matrices $A$ et $B$ par concaténation de plusieurs vecteurs (**cbind(), rbind()**):

$$A = \begin{pmatrix} 4 & -1 & 1 \\ 3 & 10 & 3 \end{pmatrix}, \quad B = \begin{pmatrix} 2 & 4 \\ 3 & 3 \\ 4 & 2 \end{pmatrix}$$

2. Sélectionner la deuxième colonne de $A$, et le coefficient $3,2$ de $B$.
3. Interpréter le résultat des instructions `A*B`, `A[,1:2]*B[2:3,]`et `M=A%*%B`.

***Correction.***

```
A=rbind(c(4,-1,1),
        c(3,10,3))
B=cbind(2:4,4:2)
```

```
A[,2]             # deuxième colonne
## [1] -1 10
B[3,2]            # troisième ligne deuxième colonne
## [1] 2
# A*B             # erreur, car les opérations sont faites composantes par composante
A[,1:2]*B[2:3,]   # OK les matrices sont de même taille
##      [,1] [,2]
## [1,]   12   -3
## [2,]   12   20
A%*%B             # le produit matriciel
##      [,1] [,2]
## [1,]    9   15
## [2,]   48   48
solve(A%*%B)%*%A%*%B  # petite vérification
##               [,1]         [,2]
## [1,]  1.000000e+00 1.665335e-16
## [2,] -1.110223e-16 1.000000e+00
```

# 2 Régression linéaire

Un ingénieur d'une entreprise de semi-conducteurs souhaite modéliser le `gain` d'un appareil électronique en fonction de trois variables [1] : résistance de l'émetteur (`emitter`), résistance de la base (`base`), résistance de l'émetteur à la base (`etr`). Il dispose de 19 relevés indépendants du gain pour des valeurs différentes des trois caractéristiques.

## 2.1 Acquérir les données

Les données sont lues dans un dataframe à partir du fichier `hFE.csv`.

```
rm(list=objects()); graphics.off()
# setwd("~/TD") # à personnaliser
df=read.table("hFE.csv",sep=";",dec=",",header=TRUE)
```

1. Quelle est l'utilité de chaque argument utilisé dans cet appel à la fonction `read.table`? Valider la commande de lecture en commentant les résultats des commandes suivantes

***Correction.*** *Leq arguments* `sep`, `dec` *et* `header` *sont définis car pas appelés avec les valeurs par défaut:* `sep` *indique que le séparateur par défaut est le point-virgule,* `dec` *que la décimale est un point (typique d'un codage français), et* `header` *que la première ligne du fichier contient le nom des variables.*

*On vérifie qu'on a bien lu $n = 19$ observations et 4 variables (*`dim`*) quantitatives (*`str`*).*

```
head(df)
##     gain emitter  base emitter.to.recepteur
## 1 128.40   14.62 226.0                7.000
## 2  52.62   15.63 220.0                3.375
## 3 113.90   14.62 217.4                6.375
## 4  98.01   15.00 220.0                6.000
## 5 139.90   14.50 226.5                7.625
## 6 102.60   15.25 224.1                6.000
dim(df)
```

1. R. Myers, D. Montgomery, G. Vining: Generalized linear models (Wiley 2002)

```
## [1] 19  4
str(df)
## 'data.frame':    19 obs. of  4 variables:
##  $ gain                 : num  128.4 52.6 113.9 98 139.9 ...
##  $ emitter              : num  14.6 15.6 14.6 15 14.5 ...
##  $ base                 : num  226 220 217 220 226 ...
##  $ emitter.to.recepteur: num  7 3.38 6.38 6 7.62 ...
```

2. Renommer la quatrième variable avec un nom plus commode, vérifier le résultat

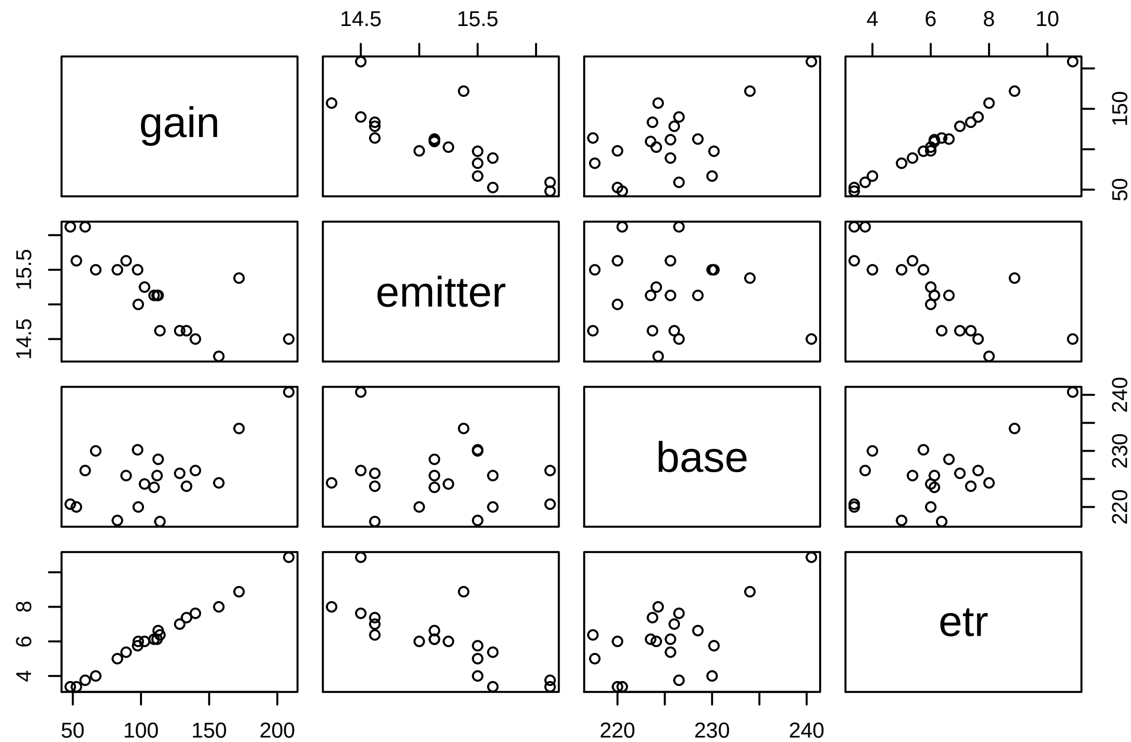
```
names(df)[4]='etr'
```

```
str(df)
## 'data.frame':    19 obs. of  4 variables:
##  $ gain   : num  128.4 52.6 113.9 98 139.9 ...
##  $ emitter: num  14.6 15.6 14.6 15 14.5 ...
##  $ base   : num  226 220 217 220 226 ...
##  $ etr    : num  7 3.38 6.38 6 7.62 ...
```

### 2.2 Analyse uni- et bi-variée

Commenter l'analyse univariée et bivariée.

```
summary(df)
##       gain           emitter            base            etr
##  Min.   : 48.14   Min.   :14.25   Min.   :217.4   Min.   : 3.375
##  1st Qu.: 85.89   1st Qu.:14.62   1st Qu.:222.0   1st Qu.: 5.188
##  Median :109.60   Median :15.13   Median :225.6   Median : 6.125
##  Mean   :109.66   Mean   :15.16   Mean   :225.5   Mean   : 6.191
##  3rd Qu.:130.90   3rd Qu.:15.50   3rd Qu.:227.5   3rd Qu.: 7.188
##  Max.   :208.40   Max.   :16.12   Max.   :240.5   Max.   :10.870
pairs(df)
```




```
round(cor(df),3)
##              gain emitter   base    etr
## gain        1.000  -0.772  0.603  0.995
## emitter    -0.772   1.000 -0.140 -0.768
## base        0.603  -0.140  1.000  0.598
## etr         0.995  -0.768  0.598  1.000
```

```
library(corrplot)
corrplot(cor(df))

library(GGally)
ggpairs(df)
```

***Correction.*** *On observe une très forte corrélation positive entre gain et etr, une forte corrélation négative entre gain et emitter. Les variables explicatives sont elles-mêmes corrélées entre elles.*

## 2.3 Modéliser

Après cette étude descriptive préliminaire, quel modèle proposer?

***Correction.*** *Modèle de régression linéaire*

$$gain_i = \mu + \beta_1\ emitter_i + \beta_2 base_i + \beta_3 etr_i + \varepsilon_i$$

*où le bruit* $\varepsilon \sim \mathcal{N}(0, \sigma^2 Id_n)$. $\mu$ *est le paramètre d'intercept et le paramètre à estimer est* $\theta = (\mu, \beta_1, \beta_2, \beta_3)$ *de dimension* $p = 4$ *qui est également celle du modèle. La variance* $\sigma^2$ *joue le rôle d'un paramètre de nuisance qu'il faudra aussi estimer.*

## 2.4 Estimer: calcul "à la main"

1. Identifier dans le code suivant les éléments vus dans le cours pour l'estimation d'une régression linéaire.

```
X=as.matrix(df)
X[,1]=1
Y=as.matrix(df$gain)
theta.est=solve(t(X)%*%X)%*% t(X)%*%Y
n=length(Y)
p=ncol(X)
sigma.est=sqrt(sum( (X%*%theta.est -Y)^2   )/(n-p))
V= solve(t(X)%*%X) *sigma.est^2
stddev=sqrt(diag(V))
```

***Correction.*** *$X$ est la matrice du plan d'expérience, la première colonne ne contient que des 1, c'est l'intercept, les autres colonnes sont les variables explicatives. $Y$ est le gain, variable à expliquer. D'après le cours*

- $\widehat{\theta} = (X'X)^{-1}X'Y$ *(calculé dans* `theta.est`*)*
- $\widehat{\sigma}^2 = ||Y - X\widehat{\theta}||^2/(n-p)$ *(calculé dans* `sigma.est`*),*
- *la matrice de variance estimée* `V` *de $\widehat{\theta}$ est $\widehat{\sigma}^2(X'X)^{-1}$*
- *l'écart type de la variance estimée de la composante $\theta_j$ du paramètre est $\sqrt{V_{jj}}$.*

2. Quelle est l'équation de l'estimée de la fonction de régression? Calculer l'estimation ponctuelle du gain pour les valeurs de covariables de la deuxième observation, puis pour l'ensemble des observations.

***Correction.*** *La fonction de régression est*

$$\widehat{y}(emitter, base, etr) = \widehat{\mu} + \widehat{\beta}_1\ emitter + \widehat{\beta}\ base + \widehat{\beta}_3\ etr = (1\ emitter\ base\ etr)\ \widehat{\theta}$$

*avec les valeurs des coefficients estimés par le logiciel.*

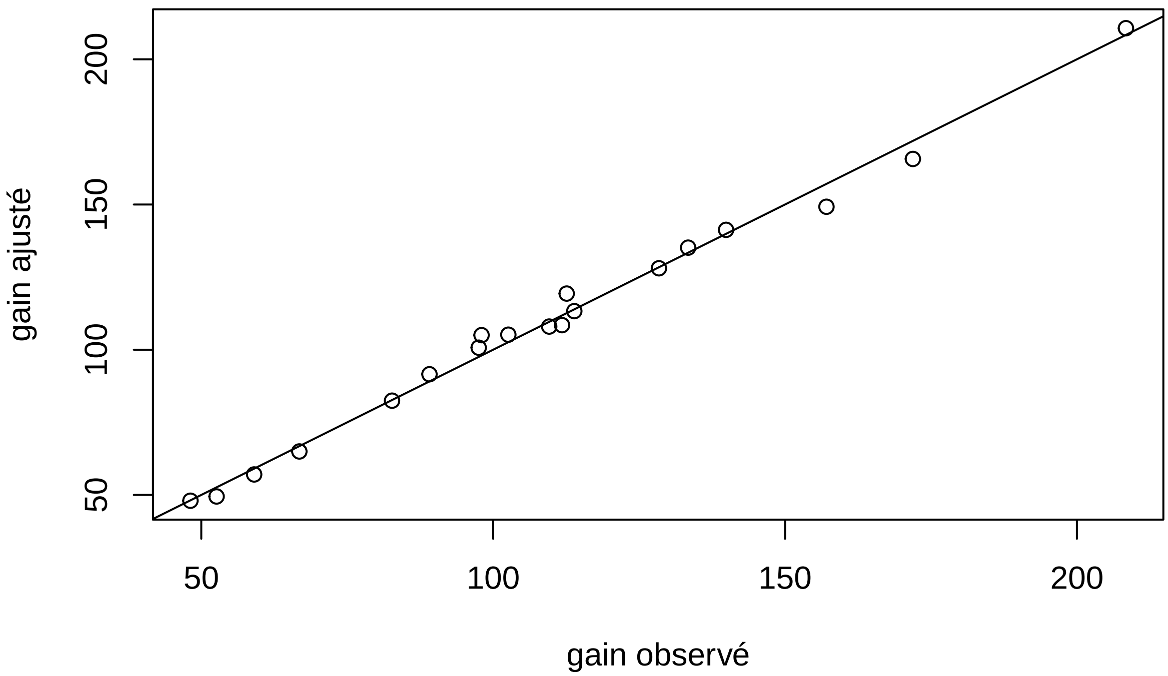
```
# pour la deuxième observation
X[2,]%*%theta.est
##          [,1]
## [1,] 49.46775

# pour toutes les observations
Yest=X%*%theta.est
head(Yest)
##           [,1]
## [1,] 128.03671
## [2,]  49.46775
## [3,] 113.28563
## [4,] 105.02137
## [5,] 141.28377
## [6,] 105.17249
```

## 2.5 Représentation

Tracer le gain ajusté en fonction du gain observé et commenter

***Correction.*** *Le modèle est très bien ajusté, comme le montre visuellement la figure ci-dessous: les points s'allongent autour de la première bissectrice, les ajustements très proches des valeurs observées.*

```
plot(df$gain,X%*%theta.est,xlab="gain observé", ylab="gain ajusté")
abline(0,1)       # la première bissectrice
```




### 2.6 Estimer: fonction ‘lm‘

Le modèle linéaire peut être directement estimé par la fonction `lm` (linear model), dont le premier argument indique le modèle sous la forme `y~formule` où `y` est le nom de la variable réponse et `formule` les noms des variables explicatives concaténés par des `+`. On peut remplacer `formule` par `.` pour mentionner que toutes les variables du dataframe (hormis la réponse) sont explicatives.

1. Faire la correspondance entre les informations calculées à la main et celles rendues par `summary`. Accéder à ces informations en exploitant l'objet sorti par la fonction `lm`. Identifier également les informations contenues dans l'objet en sortie de `summary`.

```
## lm(gain~emitter+base+etr,data=df)
model3=lm(gain~.,data=df)
summary(model3)
##
## Call:
## lm(formula = gain ~ ., data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -7.0114 -2.3900  0.2204  1.9219  7.8777
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -20.3129    51.8246  -0.392    0.701
## emitter      -3.2241     3.6100  -0.893    0.386
## base          0.2334     0.2768   0.843    0.412
## etr          20.3895     1.2662  16.103  7.1e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.163 on 15 degrees of freedom
## Multiple R-squared:  0.9915, Adjusted R-squared:  0.9898
## F-statistic: 584.6 on 3 and 15 DF,  p-value: 9.386e-16
names(model3)     # le nom des informations sorties par summary
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"
model3$coef       # accès à l'une de ces informations
## (Intercept)     emitter        base         etr
##  -20.312904   -3.224073    0.233447   20.389508
## à compléter !
```

***Correction.*** *On fait les différentes correspondances ci-dessous*

```
# colonne Estimate: valeur estimée du paramètre
theta= model3$coef; theta
## (Intercept)     emitter        base         etr
##  -20.312904   -3.224073    0.233447   20.389508

# Std Error: racine carrée de la variance de l'estimation de chaque composante
V=vcov(model3);V      # variance de l'estimateur
##              (Intercept)     emitter        base        etr
## (Intercept) 2685.793642 -81.7839590 -6.17232437 -8.5275598
## emitter      -81.783959  13.0321000 -0.62192724  3.9415365
## base          -6.172324  -0.6219272  0.07664277 -0.2712626
## etr           -8.527560   3.9415365 -0.27126260  1.6032420
s=sqrt(diag(V));s      # colonne Std. Error
## (Intercept)     emitter        base         etr
##  51.8246432   3.6100000   0.2768443   1.2661919

# estimation de sigma
sqrt(sum((df$gain-model3$fitted)^2)/model3$df.residual) #4.163142
## [1] 4.163142

# la dimension de l'espace des résidus
model3$df.residual
## [1] 15

# valeurs ajustées
model3$fitted
##         1         2         3         4         5         6         7
## 128.03671  49.46775 113.28563 105.02137 141.28377 105.17249  48.00468
##         8         9        10        11        12        13        14
```

```
## 107.96800  82.45956 119.32999 100.69312  57.05142 108.45824  91.55407
##        15        16        17        18        19
## 165.68432  64.96479 149.22227 210.71598 135.14585

# et l'objet en sortie de summary
outsum=summary(model3)
names(outsum)
##  [1] "call"          "terms"         "residuals"     "coefficients"
##  [5] "aliased"       "sigma"         "df"            "r.squared"
##  [9] "adj.r.squared" "fstatistic"    "cov.unscaled"
outsum$sigma #une autre façon d'accéder à l'estimation de sigma
## [1] 4.163142
```

***Correction.*** *(suite)*

*On peut aussi interpréter les colonnes* `t value` *(valeur de la statistique de student observée $t_{obs}$ du test $\theta_j = 0$ contre $\theta_j \neq 0$) et* `Pr(>|t|)` *(p-value de ce test).*

*La statistique de test est $T_n = \widehat{\theta}_j/\widehat{\sigma}_j$ où $\widehat{\sigma}_j = \widehat{V}_{jj}$ qui suit une loi $\mathcal{T}_{n-p}$ sous $(H_0)$. La région de rejet de viveau $\alpha$ est $\mathcal{R} = \{|T_n| > q^*_{1-\alpha/2}\}$. La pvalue vaut $2\mathbb{P}(|T_n| > |t_{obs}|)$.*

*Par exemple pour le test de niveau $\alpha = 5\%$ de $\theta_{emitter} = 0$ contre $\theta_{emitter} \neq 0$ on ne peut rejeter $(H_0)$ ($|t_obs| < q^*_{1-\alpha/2}$ ou $p - value > \alpha$) qu'on accepte: emitter n'est pas significatif, décision prise avec risque de seconde espèce inconnu. Il n'est pas utile de retenir emitter dans le modèle.*

*En revanche, le $\theta_{etr} = 0$ contre $\theta_{etr} \neq 0$ rejette $(H_0)$ $((|t_obs| > q^*_{1-\alpha/2}$ ou $p - value < \alpha))$: emitter est (hautement) significatif, le risque de la décision est de 1ère espèce $\alpha = 5\%$.*

```
# t value: valeur de la statistique de student du test theta_j=0 contre theta_j<>0
tobs=model3$coeff/s

# colonne Pr(>|t|): p-value du test précédent
alpha=0.05
q=qt(1-alpha/2,model3$df) ;q
## [1] 2.13145

abs(tobs)>q
## (Intercept)     emitter        base         etr
##       FALSE       FALSE       FALSE        TRUE

2*pt(-abs(tobs ),model3$df)
##  (Intercept)      emitter         base          etr
## 7.006043e-01 3.859108e-01 4.123380e-01 7.100883e-11
2*(1-pt(abs(tobs),model3$df) ) #idem
##  (Intercept)      emitter         base          etr
## 7.006043e-01 3.859108e-01 4.123380e-01 7.100898e-11
```

2. Comparer les résultats de 2.4 avec ceux de la fonction `predict`

```
predict(model3)
##         1         2         3         4         5         6         7
## 128.03671  49.46775 113.28563 105.02137 141.28377 105.17249  48.00468
##         8         9        10        11        12        13        14
## 107.96800  82.45956 119.32999 100.69312  57.05142 108.45824  91.55407
##        15        16        17        18        19
## 165.68432  64.96479 149.22227 210.71598 135.14585
```

***Correction.*** *On retrouve bien évidemment les mêmes résultats*

```
head( cbind(predict(model3),Yest) )
##        [,1]      [,2]
## 1 128.03671 128.03671
## 2  49.46775  49.46775
## 3 113.28563 113.28563
## 4 105.02137 105.02137
## 5 141.28377 141.28377
## 6 105.17249 105.17249
```

### 2.7 Intervalle de confiance

1. Calculer un intervalle de confiance de niveau 95% pour chacun des paramètres du modèle (`vcov`, `qt`). Comparer avec les résultats de la fonction `confint`.
2. Prédire par intervalle de confiance de niveau 95% un gain moyen quand `x1=14.5`, `x2=220`, `x3=5.0`. Retrouver ce résultat en utilisant la fonction `predict`.
3. Comparer avec un intervalle de niveau 95% pour une valeur individuelle de gain sous les mêmes conditions et commenter.

***Correction.*** *Comme $\widehat{\theta} \sim \mathcal{N}(\theta, \sigma^2(X'X)^{-1})$, on déduit pour chaque composante un IC de $\theta$ de niveau $1-\alpha$*

$$IC(\theta_j) = \left[\widehat{\theta}_j - q_{1-\alpha/2}^{\mathcal{T}_{n-p}} \frac{\widehat{\sigma}_j}{\sqrt{n}};\ \widehat{\theta}_j + q_{1-\alpha/2}^{\mathcal{T}_{n-p}} \frac{\widehat{\sigma}_j}{\sqrt{n}}\right]$$

*Soit $x = [1\ a\ b\ c]$ une nouvelle condition d'expérience. Comme $x\widehat{\theta} \sim \mathcal{N}(x\theta, \sigma^2 x(X'X)^{-1}x')$, on déduit un IC de $\mathbb{E}(Y|x) = x\theta$ de niveau $1-\alpha$*

$$IC(x\theta_j) = \left[x\widehat{\theta}_j - q_{1-\alpha/2}^{\mathcal{T}_{n-p}} \frac{\widehat{\sigma}_j^x}{\sqrt{n}};\ x\widehat{\theta}_j + q_{1-\alpha/2}^{\mathcal{T}_{n-p}} \frac{\widehat{\sigma}_j^x}{\sqrt{n}}\right] \text{ avec } \widehat{\sigma}_j^x = \widehat{\sigma}\sqrt{x(X'X)^{-1}x'}$$

*et pour une valeur individuelle*

$$IC(\mathbb{E}(Y^{new})) = \left[x\widehat{\theta}_j - q_{1-\alpha/2}^{\mathcal{T}_{n-p}} \frac{\tilde{\sigma}_j^x}{\sqrt{n}};\ x\widehat{\theta}_j + q_{1-\alpha/2}^{\mathcal{T}_{n-p}} \frac{\tilde{\sigma}_j^x}{\sqrt{n}}\right] \text{ avec } \tilde{\sigma}_j^x = \widehat{\sigma}\sqrt{x(X'X)^{-1}x'+1}$$

```
# IC de theta
qt=qt(1-alpha/2,model3$df.residual)
IC=data.frame(mean=theta,
              min= theta-qt*sqrt(diag(V)) ,
              max= theta+qt*sqrt(diag(V)))
IC
```

```
##                    mean          min        max
## (Intercept) -20.312904 -130.7745162 90.1487082
## emitter      -3.224073  -10.9186062  4.4704595
## base          0.233447   -0.3566327  0.8235266
## etr          20.389508   17.6906840 23.0883325

confint(model3)   #idem
##                    2.5 %     97.5 %
## (Intercept) -130.7745162 90.1487082
## emitter      -10.9186062  4.4704595
## base          -0.3566327  0.8235266
## etr           17.6906840 23.0883325
# 0 appartient à l'IC de la base, celle-ci n'est pas significative
# 0 n'appartient pas à l'IC de etr, etr est significatif

### IC de l'espérance sous une condition d'expérience
predict(model3,newdata=data.frame(emitter=14.5,base=220,etr=5.),
        interval="confidence")
##       fit      lwr      upr
## 1 86.2439 79.70896 92.77885

# REM: IC pour toutes les conditions du jeu de données
predict(model3)
##         1         2         3         4         5         6         7
## 128.03671  49.46775 113.28563 105.02137 141.28377 105.17249  48.00468
##         8         9        10        11        12        13        14
## 107.96800  82.45956 119.32999 100.69312  57.05142 108.45824  91.55407
##        15        16        17        18        19
## 165.68432  64.96479 149.22227 210.71598 135.14585

# calcul direct
x0=matrix(c(1,14.5,220,5.),ncol=4,nrow=1)
y0.chap=x0%*%theta
v0.chap= x0%*%V%*%t(x0)

data.frame(mean=y0.chap, min= y0.chap-qt*sqrt(v0.chap) ,
           max= y0.chap+qt*sqrt(v0.chap)) #idem
##      mean      min      max
## 1 86.2439 79.70896 92.77885

# IC de prévision d'une valeur individuelle
# il est plus large que le précédent
predict(model3,newdata=data.frame(emitter=14.5,base=220,etr=5.),
        interval="prediction")
##       fit     lwr      upr
## 1 86.2439 75.2237 97.26411
```

## 2.8 Test de significativité globale

***Correction.*** *La fonction* `lm` *effectue également le test de significativité globale du modèle iid contre le modèle d'étude. La statistique de test de Fisher suit sous* $(H_0)$ *(modèle iid) une loi de Fisher:*

$$F = \frac{(SCR(\widehat{\theta}_{H_0}) - SCR(\widehat{\theta}_{H_1}))/(p-1)}{SCR(\widehat{\theta}_{H_1})/(n-p)} \sim \mathcal{F}(p-1=3, n-p=19-4=15)$$

*Dans le modèle iid, l'emv de l'intercept est* $\bar{Y}$*, d'où* $SCR(\widehat{\theta}_{H_0}) = \sum_i (y_i - \bar{y})^2$*; dans le modèle d'étude,* $SCR(\widehat{\theta}_{H_0}) = \sum_i (y_i - x_i\widehat{\theta})^2$ *a déjà été calculée*

```
# test de Fisher
SCR0=sum((Y-mean(Y))^2)                    # 30654
SCR1=sum((Y-model3$fitted.values)^2)       # 259.9762
SCR0-SCR1                                  # SCM=30394.02
## [1] 30394.02

((SCR0-SCR1)/3)/(SCR1/(n-p))               # 584.5539
## [1] 584.5539

anova(lm(gain~1,data=df),model3) # idem, cf p69 du poly pour l'interprétation
## Analysis of Variance Table
##
## Model 1: gain ~ 1
## Model 2: gain ~ emitter + base + etr
##   Res.Df   RSS Df Sum of Sq      F    Pr(>F)
## 1     18 30654
## 2     15   260  3     30394 584.55 9.386e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

***Correction.*** *La méthode* `anova` *calcule le tableau d'analyse de la variance (cf poly 69, avec* $q = 1$ *puisque le modèle iid est de dimension 1)*

| *Source* | *Res.Df* | *RSS* | *Df* | *Sum of Sq* | *F* | *Prob>F* |
|---|---|---|---|---|---|---|
| *model1* | $n-1$ | $SCR(\widehat{\theta}_{H_0})$ | | | | |
| *model2* | $n-p$ | $SCR(\widehat{\theta}_{H_1})$ | $p-1$ | $SCM = SCR(\widehat{\theta}_{H_0}) - SCR(\widehat{\theta}_{H_1})$ | $f_{obs} = \frac{SCM/(p-1)}{SCR(\widehat{\theta}_{H_1})/(n-p)}$ | $P(F > f_{obs})$ |

*On peut égalemet utiliser la statistique de Wald dont on dérive une statistique de loi exacte Fisher.*

$$W_{exact} = W/(p-1) = \frac{(A\widehat{\theta})'(AVA')^{-1}(A\widehat{\theta})/(p-1)}{\widehat{\sigma}^2/(n-p)} \sim \mathcal{F}(p-1, n-p) \text{ avec } A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

*On a admis dans le cours l'égalité des deux statistiques* $F = W_{exact}$*, et on vérifie que c'est le cas ici*

```
# test de significavité globale avec Wald exact ~  Fisher(3,n-p=19-4=15))
A=matrix(c(0,1,0,0,
           0,0,1,0,
           0,0,0,1),nrow=3,ncol=4,byrow=TRUE)
t(A%*%theta)%*%solve(A%*%V%*%t(A))%*% A%*%theta/3  #584.5539
##           [,1]
## [1,] 584.5539
```