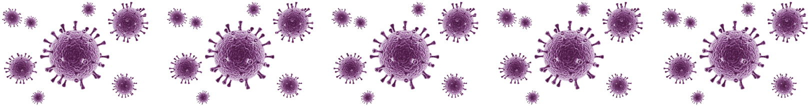
Modèles Linéaires Appliqués / Régression
Régression Logistique: Inférence

Arthur Charpentier

UQAM

Hiver 2020 - COVID-19 # 2



Modèle de régression

$$p_i = \mathbb{E}(Y_i|\boldsymbol{X}_i = \boldsymbol{x}_i) \in [0,1] \quad \neq \boldsymbol{x}_i^\top \boldsymbol{\beta}$$

$\longrightarrow$ utilisation de la côte

$$\text{odds}_i = \frac{\mathbb{P}[Y_i = 1]}{\mathbb{P}[Y_i = 0]} = \frac{p_i}{1 - p_i} \in [0, \infty].$$

soit, en passant au logarithme

$$\log(\text{odds}_i) = \log\left(\frac{p_i}{1 - p_i}\right) \in \mathbb{R}.$$

On appelle logit cette transormation,

$$\text{logit}(p_i) = \log\left(\frac{p_i}{1 - p_i}\right) = \boldsymbol{x}_i^\top \boldsymbol{\beta}$$

ou

$$p_i = \text{logit}^{-1}(\boldsymbol{x}_i^\top \boldsymbol{\beta}) = \frac{\exp[\boldsymbol{x}_i^\top \boldsymbol{\beta}]}{1 + \exp[\boldsymbol{x}_i^\top \boldsymbol{\beta}]}.$$

Maximum de Vraisemblance

La log-vraisemblance est ici

$$\log \mathcal{L}(\beta) = \sum_{i=1}^{n} y_i \log(p_i(\beta)) + (1-y_i)\log(1-p_i(\beta))$$

Conditions du premier ordre,

$$\frac{\partial \log \mathcal{L}(\beta)}{\partial \beta_k}\Big|_{\beta=\widehat{\beta}} = \sum_{i=1}^{n} \frac{y_i}{p_i(\beta)}\frac{\partial p_i(\beta)}{\partial \beta_k} - \frac{1-y_i}{p_i(\beta)}\frac{\partial p_i(\beta)}{\partial \beta_k} = 0$$

or compte tenu de la forme de $p_i(\beta)$,

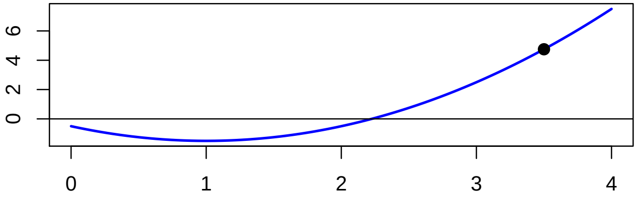
$$\frac{\partial p_i(\beta)}{\partial \beta_k} = p_i(\beta)[1-p_i(\beta)]x_{k,i}$$

on obtient

$$\frac{\partial \log \mathcal{L}(\beta)}{\partial \beta_k}\Big|_{\beta=\widehat{\beta}} = \sum_{i=1}^{n} x_{k,i}[y_i - p_i(\widehat{\beta})] = 0,\ \forall k.$$

Algorithme de Newton

On veut résoudre (numériquement) $f(x) = 0$, où $f : \mathbb{R} \to \mathbb{R}$
Commencer en x_0
Étape k: $x_k \leftarrow x_{k-1} - \dfrac{f(x_{k-1})}{f'(x_{k-1})}$



On veut résoudre (numériquement) $f(\boldsymbol{x}) = \boldsymbol{0}$, où $f : \mathbb{R}^d \to \mathbb{R}^d$
On commence avec $\boldsymbol{x}_0$
Étape k: $\boldsymbol{x}_k \leftarrow \boldsymbol{x}_{k-1} - \nabla f(\boldsymbol{x}_{k-1})^{-1} f(\boldsymbol{x}_{k-1})$

Algorithme de Newton

On veut résoudre ici $\nabla \log \mathcal{L}(\beta) = \mathbf{0}$

On commence avec β_0
Étape j: $\beta_k \leftarrow \beta^{j-1} - H(\beta^{j-1})^{-1} \nabla \log \mathcal{L}(\beta^{j-1})$

$H(\beta) = [H_{j,k}]$ est la matrice Hessienne, où

$$H_{j,k} = \frac{\partial^2 \log \mathcal{L}(\beta)}{\partial \beta_j \partial \beta_k} = -\sum_{i=1}^{n} x_{j,i} x_{k,i} p_i(\beta) [1 - p_i(\beta)]$$

soit $H(\beta) = -\boldsymbol{X}^\top \boldsymbol{\Omega} \boldsymbol{X}$ où $\boldsymbol{\Omega} = \text{diag}(\boldsymbol{p}(1 - \boldsymbol{p}))$

Posons $\boldsymbol{\Omega} = \text{diag}(\boldsymbol{p}(1-\boldsymbol{p}))$,

$$\nabla \log \mathcal{L}(\beta) = \frac{\partial \log \mathcal{L}(\beta)}{\partial \beta} = \boldsymbol{X}^\top(\boldsymbol{y}-\boldsymbol{p}) \quad (\text{avec } \boldsymbol{p} = \boldsymbol{p}(\beta))$$

$$H(\beta) = \frac{\partial^2 \log \mathcal{L}(\beta)}{\partial \beta \partial \beta^\top} = -\boldsymbol{X}^\top \boldsymbol{\Omega} \boldsymbol{X} \quad (\text{avec } \boldsymbol{\Omega} = \boldsymbol{\Omega}(\beta))$$

Algorithme de Newton:

$$\beta^j = \beta^{j-1} + \left(\boldsymbol{X}^\top \boldsymbol{\Omega}^{j-1} \boldsymbol{X}\right)^{-1} \boldsymbol{X}^\top \boldsymbol{\Omega}^{j-1}(\boldsymbol{y} - \boldsymbol{\mu}^{j-1})$$

$$\beta^j = (\boldsymbol{X}^\top \boldsymbol{\Omega} \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{\Omega} \boldsymbol{Z} \text{ où } \boldsymbol{Z} = \boldsymbol{X}\beta^{j-1} + \boldsymbol{\Omega}^{-1}(\boldsymbol{y} - \boldsymbol{p}^{j-1}),$$

qui est une régression pondérée

$$\beta^j = \text{argmin}\left\{(\boldsymbol{Z} - \boldsymbol{X}\beta)^\top \boldsymbol{\Omega}(\boldsymbol{Z} - \boldsymbol{X}\beta).\right\}$$

Maximum de Vraisemblance

Critère d'arrêt : si $\|\boldsymbol{\beta}_k - \boldsymbol{\beta}_{k-1}\| < \epsilon$, $\widehat{\boldsymbol{\beta}} = \boldsymbol{\beta}_k$

Proposition: $\widehat{\beta} \xrightarrow{\mathbb{P}} \beta$, et en posant $I(\beta) = -H(\beta) = \boldsymbol{X}^\top \boldsymbol{\Omega} \boldsymbol{X}$

$$(\widehat{\beta} - \beta) \xrightarrow{\mathcal{L}} \mathcal{N}(\mathbf{0}, I(\beta)^{-1})$$

lorsque $n \to \infty$.

Survie des Passagers du Titanic

y : indicatrice de survie d'un passager du Titanic

```
> base = base[!is.na(base$Age),]
> y = base$Survived
> X = cbind(1,(base$Sex == "male"),(base$Pclass == 2)
   ,(base$Pclass == 3),base$Age)
```

Approche par descente de gradient

```
> beta = matrix(NA,5,7)
> beta[,1]=lm(y~0+X)$coefficients
> for(s in 2:7){
eta = X%*%beta[,s-1]
p   = exp(eta)/(1+exp(eta))
Omega = diag(as.numeric(p*(1-p)),length(p),length(p))
   gradient=t(X)%*%(y-p)
   Hessian=-t(X)%*%Omega%*%X
beta[,s]=beta[,s-1]-solve(Hessian)%*%gradient }
```

Survie des Passagers du Titanic

```
> beta
        [,1]   [,2]   [,3]   [,4]   [,5]   [,6]   [,7]
[1,]   1.125  2.716  3.566  3.761  3.769  3.769  3.769
[2,]  -0.478 -2.021 -2.415 -2.510 -2.514 -2.514 -2.514
[3,]  -0.207 -0.913 -1.230 -1.298 -1.301 -1.301 -1.301
[4,]  -0.406 -1.729 -2.414 -2.566 -2.572 -2.572 -2.572
[5,]  -0.006 -0.023 -0.035 -0.037 -0.037 -0.037 -0.037
> solve(-Hessian)
         [,1]     [,2]     [,3]     [,4]     [,5]
[1,]   0.1609  -0.0372  -0.0681  -0.0881  -0.0024
[2,]  -0.0372   0.0431   0.0091   0.0146   0.0001
[3,]  -0.0681   0.0091   0.0774   0.0486   0.0007
[4,]  -0.0881   0.0146   0.0486   0.0792   0.0011
[5,]  -0.0024   0.0001   0.0007   0.0011   0.0001
```

Survie des Passagers du Titanic

Ce qui donne $\widehat{\beta}$ et la variance asymptotique (estimée) $\text{Var}(\widehat{\beta})$

Approche par moindres carrés pondérés itérés (IWLS)

```
> beta = matrix(NA,5,7)
> beta[,1]=lm(y~0+X)$coefficients
> for(s in 2:7){
eta = X%*%beta[,s-1]
p   = exp(eta)/(1+exp(eta))
Omega = diag(as.numeric(p*(1-p)),length(p),length(p))
Z   = eta + solve(Omega)%*%(y-p)
beta[,s]=lm(Z~0+X,weights=diag(Omega))$coefficients }
```

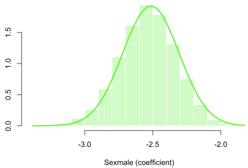
Survie des Passagers du Titanic

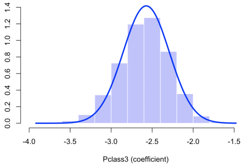
```
> beta
        [,1]   [,2]   [,3]   [,4]   [,5]   [,6]   [,7]
[1,]  1.125  2.716  3.566  3.761  3.769  3.769  3.769
[2,] -0.478 -2.021 -2.415 -2.510 -2.514 -2.514 -2.514
[3,] -0.207 -0.913 -1.230 -1.298 -1.301 -1.301 -1.301
[4,] -0.406 -1.729 -2.414 -2.566 -2.572 -2.572 -2.572
[5,] -0.006 -0.023 -0.035 -0.037 -0.037 -0.037 -0.037
> solve(t(X)%*%Omega%*%X)
         [,1]    [,2]    [,3]    [,4]    [,5]
[1,]  0.1609 -0.0372 -0.0681 -0.0881 -0.0024
[2,] -0.0372  0.0431  0.0091  0.0146  0.0001
[3,] -0.0681  0.0091  0.0774  0.0486  0.0007
[4,] -0.0881  0.0146  0.0486  0.0792  0.0011
[5,] -0.0024  0.0001  0.0007  0.0011  0.0001
```

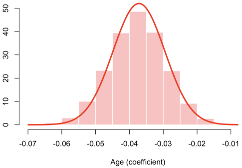
Intervalles de Confiance

L'incertitude des estimateurs peut s'appréhender par simulations (bootstrap - rééchantillonage)

```
> B = summary(glm(Survived~Sex+Pclass+Age,data=base,
    family="binomial"))$coefficients
> beta = matrix(NA,5,9999)
> n = nrow(base)
> for(b in 1:9999){
  idx = sample(1:n,size=n,replace=TRUE)
  beta[,b] = glm(Survived~Sex+Pclass+Age,data=
  base[idx,],family="binomial")$coefficients }
```

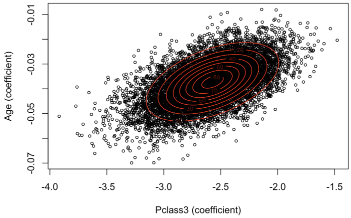


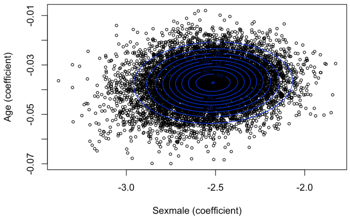




Intervalles de Confiance

```
> plot(beta[4,],beta[5,])
> m = B[4:5,1]
> V = vcov(glm(Survived~Sex+Pclass+Age,data=base,
    family="binomial"))[4:5,4:5]
> library(mnormt)
> dn = function(x,y) dmnorm(cbind(x,y),m,V)
```





donc $\widehat{\beta} \approx \mathcal{N}(\beta, \boldsymbol{\Sigma})$

```
> reg = glm(Survived~Sex+Pclass+Age,data=base,family="
    binomial")
```

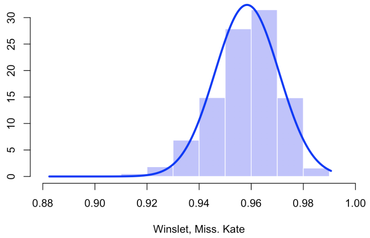
Intervalles de Confiance

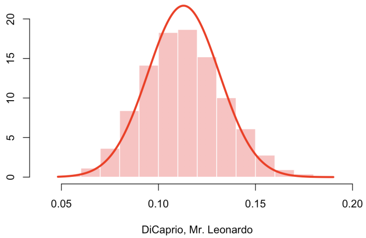
On peut regarder pour un intervalle de confiance pour $p_{\boldsymbol{x}} = \mathbb{P}(Y = 1 | \boldsymbol{X} = \boldsymbol{x})$

```
> newbase = data.frame(
  Pclass = as.factor(c(1,3)),
  Sex = as.factor(c("female","male")),
  Age = c(17,20),
  SibSp = c(1,0),
  Parch = c(2,0),
  Embarked = as.factor(c("S","S")),
  Name = as.factor(c("Winslet, Miss. Kate","DiCaprio,
    Mr. Leonardo")))
> (PRD = predict(reg,newdata=newbase,se.fit = TRUE,
    type="response"))
$fit
        1         2
0.9583891 0.1129489
$se.fit
         1          2
0.01231372 0.01840634
```

Intervalles de Confiance

```
> prd = matrix(NA,2,9999)
> for(b in 1:9999){
  idx = sample(1:n,size=n,replace=TRUE)
  regb = glm(Survived~Sex+Pclass+Age,data=base[idx,],
    family="binomial")
  prd[,b] = predict(regb,newdata=newbase,type="
    response")}
```





donc $\widehat{p}_{\boldsymbol{x}} \approx \mathcal{N}(p_{\boldsymbol{x}}, \sigma^2_{\boldsymbol{x}})$

Delta Method

$$\left(\widehat{\boldsymbol{\beta}}-\boldsymbol{\beta}\right)\xrightarrow{\mathcal{L}}\mathcal{N}\left(\mathbf{0},\boldsymbol{\Sigma}\right)$$

soit $h:\mathbb{R}^p\to\mathbb{R}^d$, différentiable, alors (Taylor)

$$h(\widehat{\beta})\approx h(\beta)+\nabla h(\beta)^\top(\widehat{\beta}-\beta)$$

si $\nabla h(\beta)\neq\mathbf{0}$, alors

$$\begin{aligned}\text{Var}\left(h(\widehat{\beta})\right)&=\text{Var}\left(h(\beta)+\nabla h(\beta)^\top(\widehat{\beta}-\beta)\right)\\&=\nabla h(\beta)^\top\Sigma\nabla h(\beta)\end{aligned}$$

et $\left(h(\widehat{\beta})-h(\beta)\right)\xrightarrow{\mathcal{L}}\mathcal{N}\left(\mathbf{0},\nabla h(\beta)^\top\boldsymbol{\Sigma}\nabla h(\beta)\right)$

Pour rappel, $\widehat{p}_{\boldsymbol{x}}=h(\boldsymbol{x}^\top\widehat{\beta})$ où $h(x)=\dfrac{e^x}{1+e^x}$.