STT3030 - Cours #0

Arthur Charpentier

Automne 2024

Lois Binomiales & Multinomiales

$Y \sim \mathcal{B}(p)$:

$$\mathbb{P}[Y=y] = p^y(1-p)^{1-y}\begin{cases} p \text{ si } y=1 \\ 1-p \text{ si } y=0 \end{cases}, \text{ où } y \in \{0,1\}$$

cf loi de Bernoulli, où $p = \mathbb{P}[Y=1] = \mathbb{E}[Y] \in [0,1]$.

$Y \sim \mathcal{B}(n,p)$:

$$\mathbb{P}[Y=y] = \binom{n}{y} p^y(1-p)^{n-y} \text{ où } y \in \{0,1,2,\dots,n\}$$

cf loi binomiale, où $\mathbb{E}[Y] = np$.

$Y_1, \cdots, Y_n$ i.i.d. $\mathcal{B}(p)$ alors $Y = \sum_{i=1}^{n} Y_i \sim \mathcal{B}(n,p)$

Lois Binomiales & Multinomiales

$\boldsymbol{Y} = (Y_1, \cdots, Y_d) \sim \mathcal{M}(\boldsymbol{p})$ où $\boldsymbol{p} = (p_1, \cdots, p_d)$ si

$$Y_1 + \cdots + Y_d = 1 \text{ et } Y_j \sim \mathcal{B}(p_j),\ \forall j \in \{1, \cdots, d\}$$

i.e. $\boldsymbol{Y} = (\mathbf{1}_{C_1}, \mathbf{1}_{C_2}, \cdots, \mathbf{1}_{C_d})$

$\boldsymbol{Y} = (Y_1, \cdots, Y_d) \sim \mathcal{M}(n, \boldsymbol{p})$ où $\boldsymbol{p} = (p_1, \cdots, p_d)$ si

$$Y_1 + \cdots + Y_d = n \text{ et } Y_j \sim \mathcal{B}(n, p_j),\ \forall j \in \{1, \cdots, d\}$$

cf loi multinomiale. Pour $(y_1, \cdots, y_d) \in \mathcal{S}_{d,n} = \{(y_1, \cdots, y_d) \in \mathbb{N}^d : (y_1 + \cdots + y_d = n\}$

$$\mathbb{P}[(Y_1, \cdots, Y_d) = (y_1, \cdots, y_d)] = \frac{n!}{y_1! \cdots y_d!} p_1^{y_1} \cdots p_d^{y_d}$$

Example: $\boldsymbol{Y} = (Y_0, Y_1) \sim \mathcal{M}(n, \boldsymbol{p})$ où $\boldsymbol{p} = (p_0, p_1)$.

Lois Binomiales : Inférence

$y_1, y_2, \ldots, y_n$ i.i.d de loi $\mathcal{B}(p)$, alors

$$\mathcal{L}(p; \boldsymbol{y}) = \prod_{i=1}^{n} \mathbb{P}(Y_i = y_i) = \prod_{i=1}^{n} p^{y_i}[1-p]^{1-y_i}$$

et la log-vraisemblance est

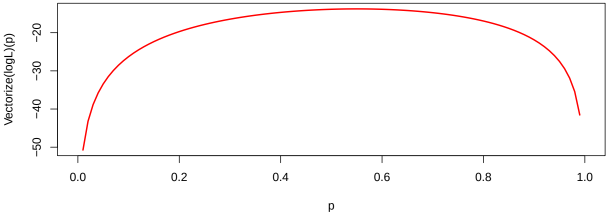
$$\log \mathcal{L}(p; \boldsymbol{y}) = \sum_{i=1}^{n} y_i \log[p] + (1-y_i)\log[1-p]$$

La condition du premier ordre est

$$\left.\frac{\partial \log \mathcal{L}(p; \boldsymbol{y})}{\partial p}\right|_{p=\widehat{p}} = \sum_{i=1}^{n} \left(\frac{y_i}{\widehat{p}} - \frac{1-y_i}{1-\widehat{p}}\right) = 0, \text{ i.e. } \widehat{p} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

Lois Binomiales : Inférence

```
> set.seed(1)
> n=20
> (Y=sample(0:1,size=n,replace=TRUE))
 [1] 0 0 1 1 0 1 1 1 1 0 0 0 1 0 1 0 1 1 0 1
> (pn = mean(Y))
[1] 0.55
> p=seq(0,1,by=.01)
> neglogL = function(p){-sum(log(dbinom(Y,1,p)))}
> plot(p,-Vectorize(neglogL)(p))
> pml = optim(fn=neglogL,par=.5,method="BFGS")$par
[1] 0.5499996
```



Lois Binomiales : Inférence

Propriété du maximum de vraisemblance

$$\sqrt{n}(p-\widehat{p}) \xrightarrow{\mathcal{L}} \mathcal{N}(0, I^{-1}(p))$$

où $I(p)$ est l'information de Fisher, i.e.

$$I(p) = -\mathbb{E}\left[\frac{\partial^2}{\partial p^2}\log f(Y,p)\right] = \frac{1}{p[1-p]}$$

$$\sqrt{n}\frac{p-\widehat{p}}{\sqrt{p(1-p)}} \xrightarrow{\mathcal{L}} \mathcal{N}(0,1) \text{ et } \sqrt{n}\frac{p-\widehat{p}}{\sqrt{\widehat{p}(1-\widehat{p})}} \xrightarrow{\mathcal{L}} \mathcal{N}(0,1)$$

d'où un intervalle de confiance approché (à 95%) pour p de la forme

$$\left[\widehat{p} \pm \frac{1.96}{\sqrt{n}}\sqrt{\widehat{p}[1-\widehat{p}]}\right].$$

Lois Binomiales : Inférence

On peut aussi construire un intervalle de confiance, par le théorème central limite, car $\widehat{p} = \overline{y}$. On sait que

$$\sqrt{n}\frac{\overline{Y} - \mathbb{E}(Y)}{\sqrt{\text{Var}(Y)}} \xrightarrow{\mathcal{L}} \mathcal{N}(0,1),$$

avec ici $\overline{Y} = \widehat{p}$, $\mathbb{E}(Y) = p$ et $\text{Var}(Y) = p(1-p)$, i.e. un intervalle de confiance est obtenu par l'approximation

$$\sqrt{n}\frac{\widehat{p} - p}{\sqrt{\widehat{p}[1-\widehat{p}]}} \xrightarrow{\mathcal{L}} \mathcal{N}(0,1),$$

d'où un intervalle de confiance (à 95%) pour p de la forme

$$\left[\widehat{p} \pm \frac{1.96}{\sqrt{n}}\sqrt{\widehat{p}[1-\widehat{p}]}\right].$$

Lois Binomiales : Inférence

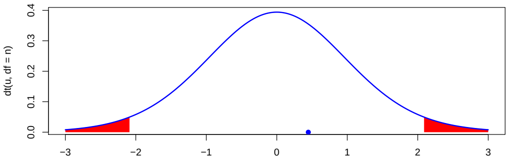
On peut faire un test de $H_0 : p = p_0$ contre $H_1 : p \neq p_0$ (par exemple 50%). On peut utiliser le test de Student,

$$T = \sqrt{n}\frac{\widehat{p} - p_0}{\sqrt{p_0(1 - p_0)}}$$

qui tend, sous H_0, vers une loi normale

```
> p0 = .5
> (T=sqrt(n)*(pn-p0)/(sqrt(p0*(1-p0))))
[1] 0.4472136
> 0.4472136<qnorm(1-alpha/2)
[1] TRUE
```

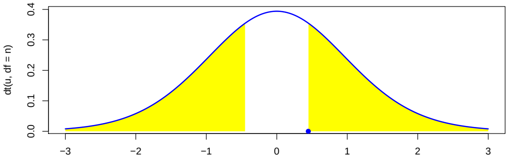
On est ici dans la région d'acceptation du test.



Lois Binomiales : Inférence

On peut aussi calculer la p-value, $\mathbb{P}(|T| > |t_{obs}|)$,

```
> 2*(1-pnorm(abs(T)))
[1] 0.6547208
```



Lois Binomiales : Inférence

Test de Wald l'idée est d'étudier la différence entre $\widehat{p}$ et p_0. Sous H_0,

$$T = n\frac{(\widehat{p} - p_0)^2}{I^{-1}(p_0)} \xrightarrow{\mathcal{L}} \chi^2(1)$$

Test du rapport de vraisemblance l'idée est d'étudier la différence entre $\log \mathcal{L}(\widehat{p})$ et $\log \mathcal{L}(p_0)$. Sous H_0,

$$T = 2\log\left(\frac{\mathcal{L}(p_0)}{\mathcal{L}(\widehat{p})}\right) \xrightarrow{\mathcal{L}} \chi^2(1)$$

Test du score l'idée est d'étudier la différence entre $\dfrac{\partial \log \mathcal{L}(p_0)}{\partial p}$ et 0. Sous H_0,

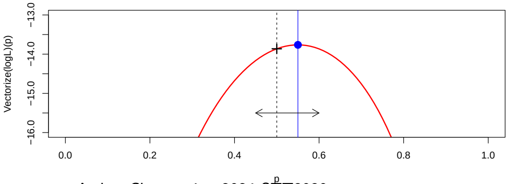
$$T = \left(\frac{1}{n}\sum_{i=1}^{n}\frac{\partial \log f_{p_0}(x_i)}{\partial p}\right)^2 \xrightarrow{\mathcal{L}} \chi^2(1)$$

Lois Binomiales : Inférence

Test de Wald différence entre $\widehat{p}$ et p_0, test de Wald. Sous H_0,

$$T = n\frac{(\widehat{p} - p_0)^2}{I^{-1}(p_0)} \xrightarrow{\mathcal{L}} \chi^2(1)$$

```
> neglogL = function(p){-sum(log(dbinom(X,1,p)))}
>  (IF = 1/(p0*(1-p0)/n))
[1] 80
>  pml=optim(fn=neglogL,par=p0,method="BFGS")$par
>  (T=(pml-p0)^2*IF)
[1] 0.199997
>  T<qchisq(1-alpha,df=1)
[1] TRUE
```

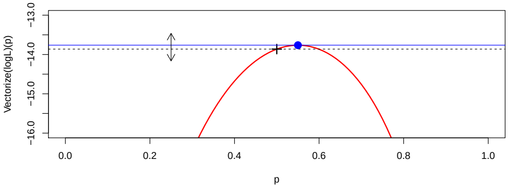


Lois Binomiales : Inférence

Test du rapport de vraisemblance l'idée est d'étudier la différence entre $\log \mathcal{L}(\widehat{p})$ et $\log \mathcal{L}(p_0)$, LRT. Sous H_0,

$$T = 2\log\left(\frac{\mathcal{L}(p_0)}{\mathcal{L}(\widehat{p})}\right) \xrightarrow{\mathcal{L}} \chi^2(1)$$

```
> logL = function(p){sum(log(dbinom(X,1,p)))}
>  (T=2*(logL(pml)-logL(p0)))
[1] 0.2003347
>  T<qchisq(1-alpha,df=1)
[1] TRUE
```

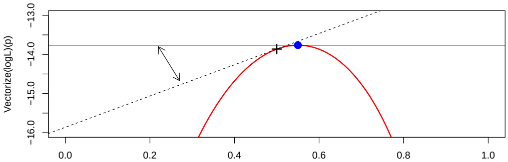


Lois Binomiales : Inférence

Test du score comparer $\dfrac{\partial \log \mathcal{L}(p_0)}{\partial p}$ et 0, test de Rao. Sous H_0

$$T = \left(\frac{1}{n}\sum_{i=1}^{n}\frac{\partial \log f_{p_0}(x_i)}{\partial p}\right)^2 \xrightarrow{\mathcal{L}} \chi^2(1)$$

```
> nx=sum(X==1)
> f = expression(nx*log(p)+(n-nx)*log(1-p))
> Df = D(f, "p")
> p=p0
> score=eval(Df)
> (T=score^2/IF)
[1] 0.2
```



Lois Multinomiales : Inférence

$\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n$ i.i.d de loi $\mathcal{M}(\boldsymbol{p})$, alors

$$\mathcal{L}(\boldsymbol{p};\boldsymbol{y}) = \prod_{i=1}^{n} \mathbb{P}(\boldsymbol{Y}_i = \boldsymbol{y}_i) = \prod_{i=1}^{n} \prod_{j=1}^{d} p_j^{y_{i,j}}$$

sous la contrainte que $\boldsymbol{p}^\top \boldsymbol{1} = 1$.
Posons $\boldsymbol{x} = (\boldsymbol{x}_{(d)}, x_d)$, i.e. $p_d = 1 - \boldsymbol{p}_{(d)}^\top \boldsymbol{1}$

$$\mathcal{L} = \prod_{i=1}^{n} \mathbb{P}(\boldsymbol{Y}_i = \boldsymbol{y}_i) = \prod_{i=1}^{n} \left(\prod_{j=1}^{d-1} p_j^{y_{i,j}} \right) \left(1 - \boldsymbol{p}_{(d)}^\top \boldsymbol{1}\right)^{\boldsymbol{y}_{i(d)}^\top \boldsymbol{1}}$$

La $j$ème condition du premier ordre est, si $s_j = \sum_{i=1}^{n} y_{i,j}$

$$\left.\frac{\partial \log \mathcal{L}}{\partial p_j}\right|_{\boldsymbol{p}=\widehat{\boldsymbol{p}}} = \frac{s_j}{\widehat{p}_j} - \frac{n - \boldsymbol{s}_{(d)}^\top \boldsymbol{1}}{1 - \widehat{\boldsymbol{p}}_{(d)}^\top \boldsymbol{1}} = 0, \text{ i.e. } \widehat{p}_j = \frac{s_j}{n}.$$

Lois Multinomiales : Inférence

L'estimateur du maximum de vraisemblance est

$$\widehat{\boldsymbol{p}} = (\widehat{p}_1, \cdots, \widehat{p}_d) = \left(\frac{s_1}{n}, \cdots, \frac{s_d}{n}\right)$$

Propriété: $\mathbb{E}(\widehat{\boldsymbol{p}}) = \boldsymbol{p}$ et $\text{Var}(\widehat{\boldsymbol{p}}) = \dfrac{1}{n}\boldsymbol{\Omega}$, où

$$\boldsymbol{\Omega} = \begin{pmatrix} p_1(1-p_1) & -p_1p_2 & \cdots & -p_1p_d \\ -p_2p_1 & p_2(1-p_2) & \cdots & -p_2p_d \\ \vdots & \vdots & \ddots & \vdots \\ -p_dp_1 & -p_dp_2 & \cdots & p_d(1-p_d) \end{pmatrix}$$

$$\sqrt{n}(\widehat{\boldsymbol{p}} - \boldsymbol{p}) \xrightarrow{\mathcal{L}} \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Omega}),$$

Remarque $\text{rang}(\boldsymbol{\Omega}) = d - 1$.

Lois Multinomiales : Inférence

Test de Pearson: H_0 : $\boldsymbol{p} = \boldsymbol{p}_0$, on utilise

$$Q = \sum_{j=1}^{d} \frac{(S_j - np_{0,j})^2}{np_{0,j}} \xrightarrow{\mathcal{L}} \chi^2(d-1),\ n \to \infty,$$

si H_0 est vraie, cf test du chi-deux.

On retrouvera ce test comme test d'indépendance.

Loi de Bernoulli : Application

y : indicatrice de survie d'un passager du Titanic

```
> loc = "http://freakonometrics.free.fr/titanic.RData"
> download.file(loc, "titanic.RData")
> load("titanic.RData")
> base = base[,1:7]
> n = nrow(base)
> (p = mean(base$Survived))
[1] 0.3838384
> p + qnorm(c(.025,.975))/sqrt(n)*sqrt(p*(1-p))
[1] 0.3519060 0.4157707
```

Si $p = \mathbb{P}(Y = 1)$, $\widehat{p} = \dfrac{342}{891} = 38.38\%$, et

$$\mathbb{P}\big(p \in [35.19\%; 41.58\%]\big) = 95\%.$$

Loi de Bernoulli : Application

$\boldsymbol{y}$: port d'embraquement des passagers du Titanic,
(Cherbourg, Queenstown, Southampton), $\boldsymbol{y} = (\mathbf{1}_C, \mathbf{1}_Q, \mathbf{1}_S)$

```
> loc = "http://freakonometrics.free.fr/titanic.RData"
> download.file(loc, "titanic.RData")
> load("titanic.RData")
> base = base[base$Embarked!="",]
> table(base$Embarked[base$Survived == 1])/sum(base$Survived == 1)

         C          Q          S
0.27352941 0.08823529 0.63823529
```

Ici $\widehat{\boldsymbol{p}} = \left(\dfrac{93}{340}, \dfrac{30}{340}, \dfrac{217}{340}\right) = (27.4\%, 8.8\%, 63.8\%)$

Modèle de régression

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

$\longrightarrow$ 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,

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

ou

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

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,

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

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

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

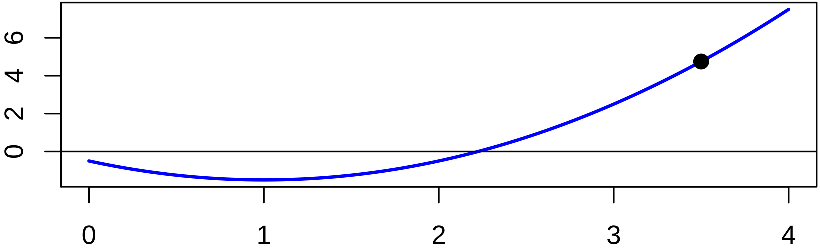
on obtient

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

Algorithme de Newton

On veut résoudre (numériquement) $f(x) = 0$, où $f : \mathbb{R} \to \mathbb{R}$

On commence en x_0, et à l'étape k: $x_k \leftarrow x_{k-1} - \dfrac{f(x_{k-1})}{f'(x_{k-1})}$



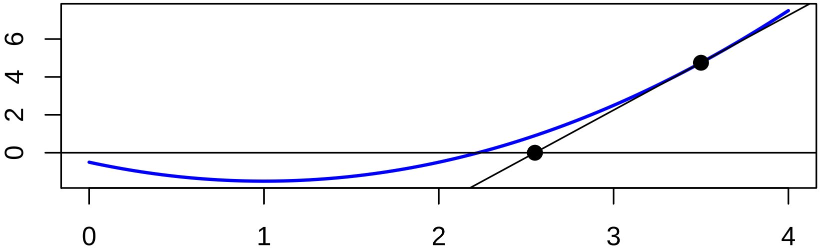
On veut résoudre (numériquement) $f(\boldsymbol{x}) = \boldsymbol{0}$, où $f : \mathbb{R}^d \to \mathbb{R}$

On commence avec $\boldsymbol{x}_0$, et à l'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 (numériquement) $f(x) = 0$, où $f : \mathbb{R} \to \mathbb{R}$

On commence en x_0, et à l'étape k: $x_k \leftarrow x_{k-1} - \dfrac{f(x_{k-1})}{f'(x_{k-1})}$



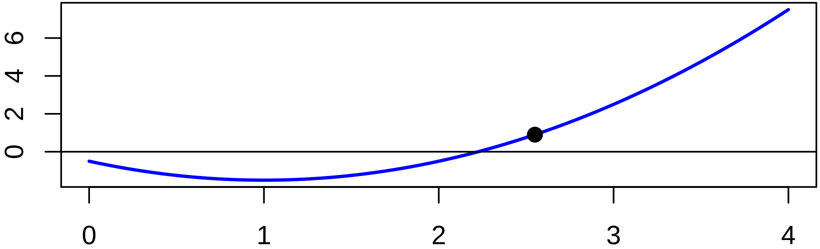
On veut résoudre (numériquement) $f(\boldsymbol{x}) = \boldsymbol{0}$, où $f : \mathbb{R}^d \to \mathbb{R}$

On commence avec $\boldsymbol{x}_0$, et à l'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 (numériquement) $f(x) = 0$, où $f : \mathbb{R} \to \mathbb{R}$

On commence en x_0, et à l'étape k: $x_k \leftarrow x_{k-1} - \dfrac{f(x_{k-1})}{f'(x_{k-1})}$



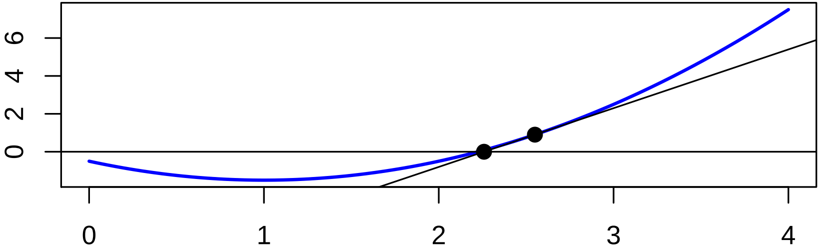
On veut résoudre (numériquement) $f(\boldsymbol{x}) = \boldsymbol{0}$, où $f : \mathbb{R}^d \to \mathbb{R}$

On commence avec $\boldsymbol{x}_0$, et à l'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 (numériquement) $f(x) = 0$, où $f : \mathbb{R} \to \mathbb{R}$

On commence en x_0, et à l'étape k: $x_k \leftarrow x_{k-1} - \dfrac{f(x_{k-1})}{f'(x_{k-1})}$



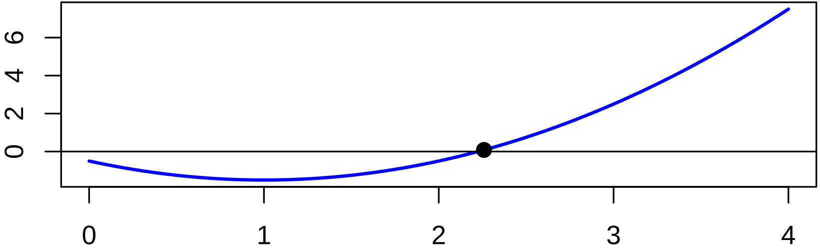
On veut résoudre (numériquement) $f(\boldsymbol{x}) = \boldsymbol{0}$, où $f : \mathbb{R}^d \to \mathbb{R}$

On commence avec $\boldsymbol{x}_0$, et à l'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 (numériquement) $f(x) = 0$, où $f : \mathbb{R} \to \mathbb{R}$

On commence en x_0, et à l'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 $f : \mathbb{R}^d \to \mathbb{R}$

On commence avec $\boldsymbol{x}_0$, et à l'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 et à l'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

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

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

Algorithme de Newton

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
}
```

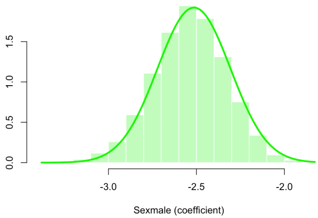
Survie des Passagers du Titanic

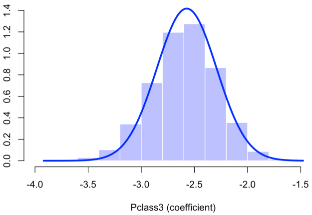
```
> beta
        [,1]   [,2]   [,3]   [,4]   [,5]   [,6]   [,7]
[1,]   1.125  2.716  3.566  3.761  3.769  3.769  3.769
[2,]  -0.478 -2.021 -2.415 -2.510 -2.514 -2.514 -2.514
[3,]  -0.207 -0.913 -1.230 -1.298 -1.301 -1.301 -1.301
[4,]  -0.406 -1.729 -2.414 -2.566 -2.572 -2.572 -2.572
[5,]  -0.006 -0.023 -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
```

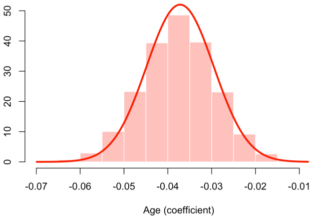
Intervalles de Confiance

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 }
```

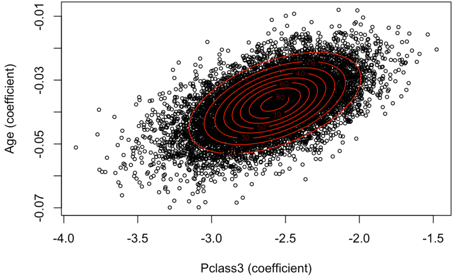


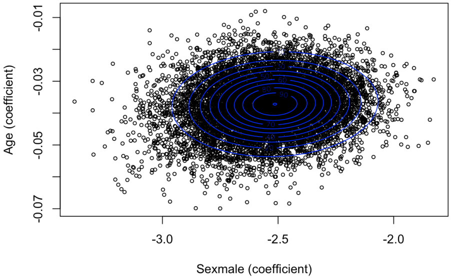




Intervalles de Confiance

```
> 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")
```

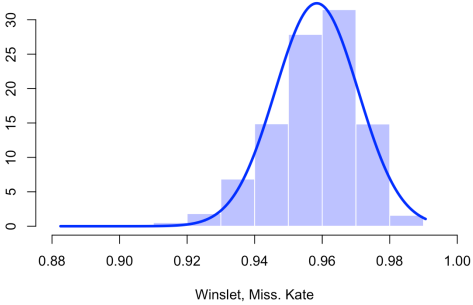
Intervalles de Confiance

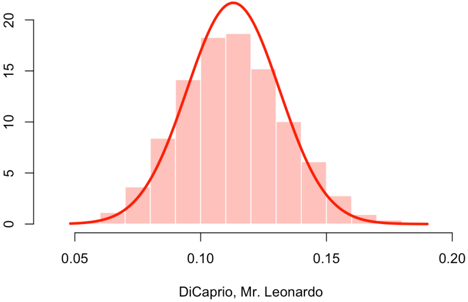
On peut regarder pour un intervalle de confiance pour $p_{\boldsymbol{x}} = \mathbb{P}(Y = 1|\boldsymbol{X} = \boldsymbol{x})$

```
> 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

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

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

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

Régression Bernoulli $y = \mathbf{1}_A$

```
> reg1 = glm((Survived==1)~Pclass+Sex+Age+I(Age^2)+I(Age^3)+SibSp,
    family=binomial, data=base)
> summary(reg1)

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  5.616e+00  6.565e-01   8.554  < 2e-16 ***
Pclass2     -1.360e+00  2.842e-01  -4.786  1.7e-06 ***
Pclass3     -2.557e+00  2.853e-01  -8.962  < 2e-16 ***
Sexmale     -2.658e+00  2.176e-01 -12.216  < 2e-16 ***
Age         -1.905e-01  5.528e-02  -3.446 0.000569 ***
I(Age^2)     4.290e-03  1.854e-03   2.314 0.020669 *
I(Age^3)    -3.520e-05  1.843e-05  -1.910 0.056188 .
SibSp       -5.041e-01  1.317e-01  -3.828 0.000129 ***
> predict(reg1)[1]
-2.592995
> predict(reg1,type="response")[1]
0.06959063
```

Régression Bernouilli $y = \mathbf{1}_{A^C}$

```
> reg0 = glm((Survived==0)~Pclass+Sex+Age+I(Age^2)+I(Age^3)+SibSp, family
    =binomial, data=base)
> summary(reg0)

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -5.616e+00  6.565e-01  -8.554  < 2e-16 ***
Pclass2      1.360e+00  2.842e-01   4.786  1.7e-06 ***
Pclass3      2.557e+00  2.853e-01   8.962  < 2e-16 ***
Sexmale      2.658e+00  2.176e-01  12.216  < 2e-16 ***
Age          1.905e-01  5.528e-02   3.446 0.000569 ***
I(Age^2)    -4.290e-03  1.854e-03  -2.314 0.020669 *
I(Age^3)     3.520e-05  1.843e-05   1.910 0.056188 .
SibSp        5.041e-01  1.317e-01   3.828 0.000129 ***

> predict(reg0)[1]
2.592995
> predict(reg0,type="response")[1]
0.9304094
```

Régression Binomiale

Au lieu de $Y_i \sim \mathcal{B}(p_i)$, $Y_i \sim \mathcal{B}(n_i, p_i)$ où n_i est connnue.

$$\mathbb{E}\left(\frac{Y_i}{n_i}\right) = p_i = \frac{e^{\boldsymbol{x}_i^\top \beta}}{1 + e^{\boldsymbol{x}_i^\top \beta}}$$

```
> reg = glm(cbind(cbind(Y,n-Y) ~ X1+X2, data = base, family=binomial)
```

On pose $z_i = y_i / n_i$, dont la densité est

$$f(y_i, p_i) = \binom{n_i}{n_i y_i} \exp\left[n_i y_i \log\left(\frac{p}{1-p}\right) + n_i \log(1-p)\right]$$

et on estime β par maximum de vraisemblance

Régression Multinomiale

Pour une loi de Bernoulli, $y \in \{0,1\}$,

$$\mathbb{P}(Y=1) = \frac{e^{\boldsymbol{x}^\top\beta}}{1+e^{\boldsymbol{x}^\top\beta}} = \frac{p_1}{p_0+p_1} \text{ et } \mathbb{P}(Y=0) = \frac{1}{1+e^{\boldsymbol{x}^\top}} = \frac{p_0}{p_0+p_1}$$

Pour une loi multinomiale, $y \in \{A,B,C\}$, $\boldsymbol{y} = (\mathbf{1}_A, \mathbf{1}_B, \mathbf{1}_C)$

$$\mathbb{P}(Y=A) = \frac{p_A}{p_A+p_B+p_C} \propto p_A \text{ i.e. } \mathbb{P}(Y=A) = \frac{e^{\boldsymbol{x}^\top\beta_A}}{e^{\boldsymbol{x}^\top\beta_B}+e^{\boldsymbol{x}^\top\beta_B}+1}$$

$$\mathbb{P}(Y=B) = \frac{p_B}{p_A+p_B+p_C} \propto p_B \text{ i.e. } \mathbb{P}(Y=B) = \frac{e^{\boldsymbol{x}^\top\beta_B}}{e^{\boldsymbol{x}^\top\beta_A}+e^{\boldsymbol{x}^\top\beta_B}+1}$$

$$\mathbb{P}(Y=C) = \frac{p_C}{p_A+p_B+p_C} \propto p_C \text{ i.e. } \mathbb{P}(Y=C) = \frac{1}{e^{\boldsymbol{x}^\top\beta_A}+e^{\boldsymbol{x}^\top\beta_B}+1}$$

Régression Multinomiale

On va essayer de comprendre la classe $y \in \{1, 2, 3\}$ sur les données du Titanic

```
> loc = "http://freakonometrics.free.fr/titanic.RData"
> download.file(loc_fichier, "titanic.RData")
> load("titanic.RData")
> regclass = multinom(Pclass ~ Sex+Age+SibSp, base)
> regclass

Coefficients:
  (Intercept)   Sexmale         Age      SibSp
2    1.416426 0.2662196 -0.04526865 -0.2150871
3    2.420469 1.0330840 -0.07541502 -0.1149161

Residual Deviance: 1347.672
AIC: 1363.672
```

Régression Multinomiale

```
> (b = coefficients(regclass))
  (Intercept)    Sexmale         Age       SibSp
2    1.416426 0.2662196 -0.04526865 -0.2150871
3    2.420469 1.0330840 -0.07541502 -0.1149161
```

Avec ici β_2 et β_3 (la class 1 est la référence)

```
> newbase = data.frame(Sex="female",Age =60,SibSp=0)
> predict(regclass,newdata=newbase,"probs")
         1          2          3
0.71708728 0.19548915 0.08742357
```

Idée : comme la class 1 est la référence,

$$\mathbb{P}(Y=1) \propto 1,\ \mathbb{P}(Y=2) \propto e^{\boldsymbol{x}^\top \boldsymbol{\beta}_2} \text{ et } \mathbb{P}(Y=3) \propto e^{\boldsymbol{x}^\top \boldsymbol{\beta}_3}$$

Régression Multinomiale

$$\mathbb{P}(Y=1) \propto 1,\ \mathbb{P}(Y=2) \propto e^{\boldsymbol{x}^\top \boldsymbol{\beta}_2} \text{ et } \mathbb{P}(Y=3) \propto e^{\boldsymbol{x}^\top \boldsymbol{\beta}_3}$$

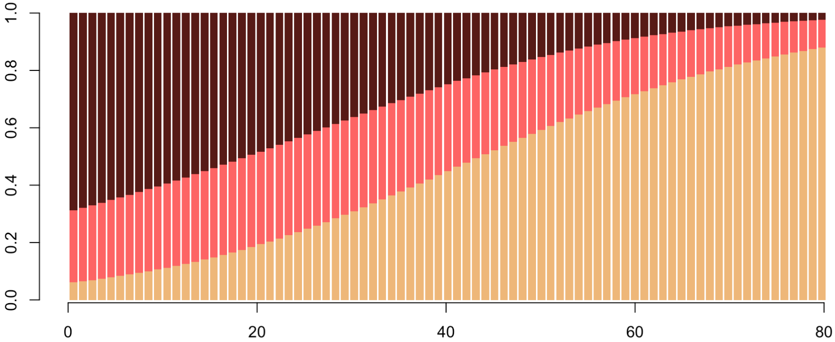
```
> x = c(1,0,60,0)
> b = rbind(rep(0,ncol(b)),b)
> t(exp(b%*%x))
     1         2         3
[1,] 1 0.2726156 0.1219148
```

$$\mathbb{P}(Y=1)\frac{1}{e^{\boldsymbol{x}^\top \boldsymbol{\beta}_2}+e^{\boldsymbol{x}^\top \boldsymbol{\beta}_3}+1},\ \mathbb{P}(Y=2)=\frac{e^{\boldsymbol{x}^\top \boldsymbol{\beta}_A}}{e^{\boldsymbol{x}^\top \boldsymbol{\beta}_B}+e^{\boldsymbol{x}^\top \boldsymbol{\beta}_B}+1},\cdots$$

```
> t(exp(b%*%x))/sum(exp(b%*%x))
             1         2          3
[1,] 0.7170873 0.1954892 0.08742357
```

Régression Multinomiale

```
> x = cbind(1,0,0:80,0)
> p2 = exp(apply((x%*%b[2,]),1,sum))
> p3 = exp(apply((x%*%b[3,]),1,sum))
> pp2 = p2/(1+p2+p3)
> pp3 = p3/(1+p2+p3)
> p = rbind(1-pp2-pp3,pp2,pp3)
> barplot(p)
```



Régression Multinomiale

Considérons une approche alternative : régressions Bernoulli itérées
considérons un premier modèle de Bernoulli $y_1 = \mathbf{1}_A$

```
> reg1 = glm((Pclass==1) ~ Sex+Age+SibSp, base, family=binomial)
```

considérons un premier modèle de Bernoulli $y_2 = \mathbf{1}_B$, entre les classes B et C

```
> reg2 = glm((Pclass==2) ~ Sex+Age+SibSp, base, family=binomial,subset =
    (Pclass!=1))
```

Idée : $\mathbb{P}(y = B) = \mathbb{P}(y = B | y \neq A) \cdot \mathbb{P}(y \neq A)$

```
> p11 = predict (reg1, newdata=base, type="response")
> p12 = predict (reg2, newdata=base, type="response")
> itp = cbind(p11,(1-p11)*p12,(1-p11)*(1-p12))
```

Régression Multinomiale

On peut comparer les modèles logit itérés (à gauche)
et le modèle multinomial (à droite)

```
> mmp = predict(regclass,newdata=base,"probs")
> head(cbind(itp,mmp))
      1     2     3       1     2     3
1 0.129 0.204 0.668   0.126 0.201 0.673
2 0.459 0.274 0.267   0.462 0.275 0.264
3 0.256 0.328 0.416   0.259 0.330 0.411
4 0.412 0.285 0.303   0.417 0.284 0.299
5 0.227 0.253 0.521   0.229 0.253 0.518
```

Références

Hilbe, J. (2009). Logistic regression models.

Hilbe, J. M. (2016). *Practical guide to logistic regression*. CRC Press.

Hosmer Jr, D. W., Lemeshow, S., and Sturdivant, R. X. (2013). *Applied logistic regression*. John Wiley & Sons.

James, G., Witten, D., Hastie, T., Tibshirani, R., et al. (2013). *An introduction to statistical learning*, volume 112. Springer.