Actuariat IARD - ACT2040
Partie 2 - régression logistique et arbres de régression ($Y \in \{0, 1\}$)

Arthur Charpentier

charpentier.arthur@uqam.ca

http ://freakonometrics.hypotheses.org/

UQÀM
Université du Québec à Montréal

Automne 2013

Modèlisation d'une variable dichotomique

Références : Frees (2010), chapitre 11 (p 305-342), Greene (2012), sections 17.2 et 17.3 (p 863-715), de Jong & Heller (2008), chapitre 7

Remarque : la régression logistique est un cas particulier des modèles GLM, avec une loi binomiale et une fonction de lien logit.

```
logit = function(formula, lien="logit", data=NULL) {
glm(formula,family=binomial(link=lien),data) }
```

Modèles introduits par Gaddum (1933), Bliss (1935), ou Berkson (1944).

La loi binomiale $\mathcal{B}(\pi)$

Soient Y_i des variables aléatoires prenant les valeurs $y_i \in \{0, 1\}$, avec probabilité $1 - \pi_i$ et π_i (respectivement), i.e.

$$\mathbb{P}(Y_i = y_i) = \pi_i^{y_i}[1 - \pi_i]^{1-y_i}, \text{ pour } y_i \in \{0, 1\}.$$

avec $\pi_i \in [0, 1]$. Aussi, $\mathbb{P}(Y_i = 1) = \pi_i$ et $\mathbb{P}(Y_i = 0) = 1 - \pi_i$. Alors $\mathbb{E}(Y_i) = \pi_i$ et $\text{Var}(Y_i) = \pi[1 - \pi_i]$.

En statistique, on suppose que $\pi_i = \pi \ \forall i \in \{1, 2, \cdots, n\}$, et on cherche à estimer π

Maximum de vraisemblance et méthode des moments

La fonction de vraisemblance, pour un échantillon $y_1, \cdots, y_n$ s'écrit

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

et la log-vraisemblance est

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

La condition du premier ordre est

$$\frac{\partial \log \mathcal{L}(\pi; \boldsymbol{y})}{\partial \pi} = \sum_{i=1}^{n} \frac{y_i}{\pi} - \frac{1-y_i}{1-\pi} = 0$$

Intervalle de confiance pour π

On a un intervalle de confiance asymptotique car on sait que l'estimateur de vraisemblance est asymptotiquement efficace,

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

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

$$I(\pi) = \frac{1}{\pi[1-\pi]}$$

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

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

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

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

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

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

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

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

Avec la méthode du score

$$\sqrt{n}\left|\frac{\widehat{\pi}-\pi}{\pi[1-\pi]}\right| \leq 1.96$$

i.e.(en élevant au carré)

$$\frac{2n\widehat{\pi}+1.96^2}{2(n+1.96^2)} \pm \frac{1.96\sqrt{1.96^2+4n\widehat{\pi}[1-\widehat{\pi}]}}{2(n+1.96^2)}$$

Mise en oeuvre pratique

Considérons l'échantillon suivant, $\{y_1, \cdots, y_n\}$

```
> 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
```

Comme $Y_i \sim \mathcal{B}(\pi)$, avec $\pi = \mathbb{E}(Y)$, l'estimateur de la méthde des moments est $\widehat{\pi} = \overline{y}$, soit ici

```
> mean(X)
[1] 0.55
```

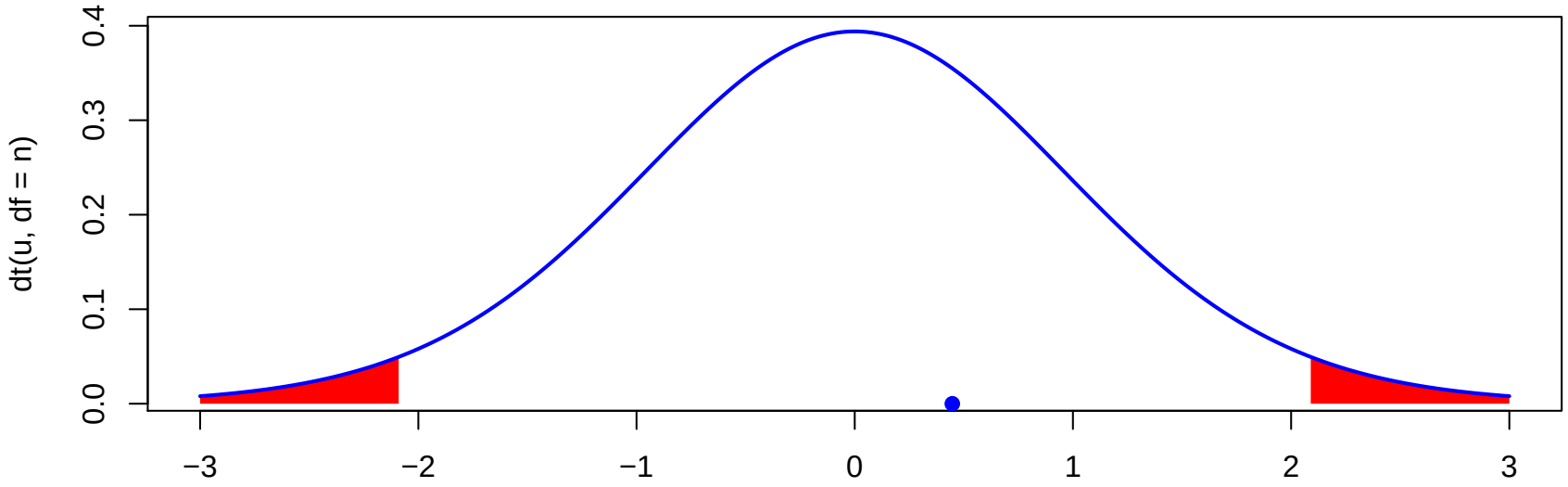
On peut faire un test de $H_0 : \pi = \pi_\star$ contre $H_1 : \pi \neq \pi_\star$ (par exemple 50%)

On peut utiliser le test de Student,

$$T = \sqrt{n}\frac{\widehat{\pi} - \pi_\star}{\sqrt{\pi_\star(1-\pi_\star)}}$$

qui suit, sous H_0 une loi de Student à n degrés de liberté.

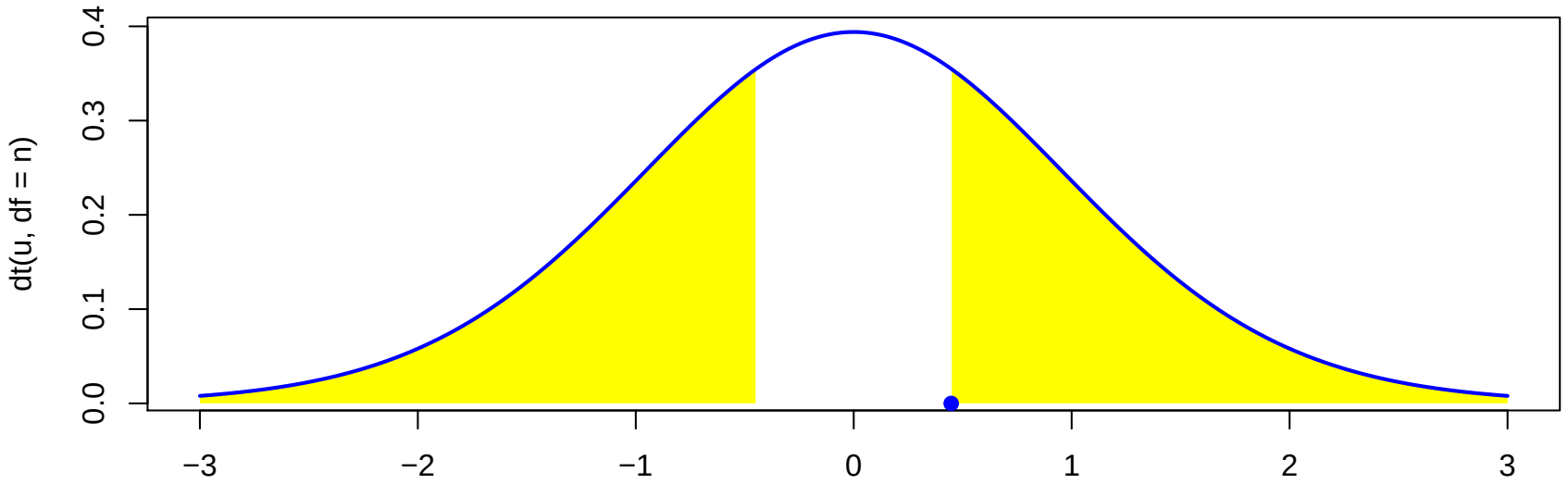
```
> (T=sqrt(n)*(pn-p0)/(sqrt(p0*(1-p0))))
[1] 0.4472136
> abs(T)<qt(1-alpha/2,df=n)
[1] TRUE
```



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

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

```
>  2*(1-pt(abs(T),df=n))
[1] 0.6595265
```



Le maximum de vraisemblance est important car il permet de faire des tests statistiques.

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

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

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

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

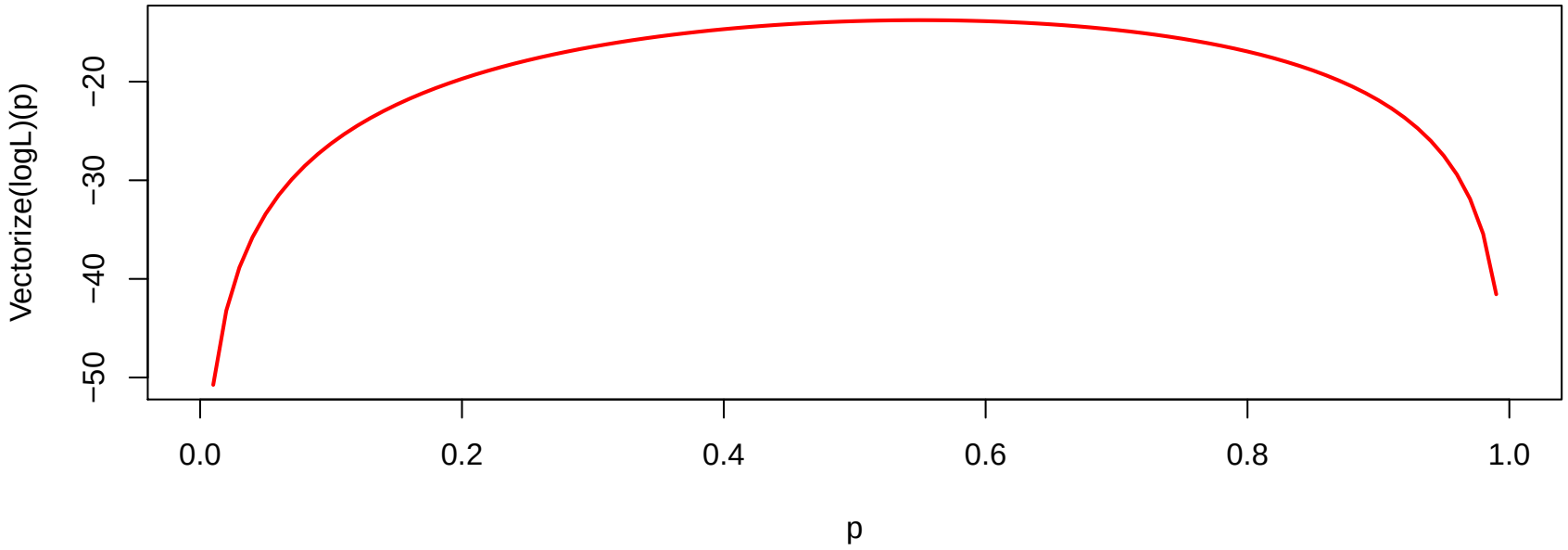
Test du score l'idée est d'étudier la différence entre $\dfrac{\partial \log \mathcal{L}(\pi_\star)}{\partial \pi}$ et 0. Sous H_0,

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

Mise en oeuvre pratique

Pour l'estimateur de maximum de vraisemblance, on peut faire les calculs, ou le faire numériquement. On commence par tracer la fonction $\theta \mapsto \log \mathcal{L}(\theta)$

```
> p=seq(0,1,by=.01)
> logL=function(p){sum(log(dbinom(X,size=1,prob=p)))}
> plot(p,Vectorize(logL)(p),type="l",col="red",lwd=2)
```



On peut trouver numériquement le maximum de $\log \mathcal{L}$,

```
> neglogL=function(p){-sum(log(dbinom(X,size=1,prob=p)))}
> pml=optim(fn=neglogL,par=p0,method="BFGS")
> pml
$par
[1] 0.5499996
```

```
$value
[1] 13.76278
```

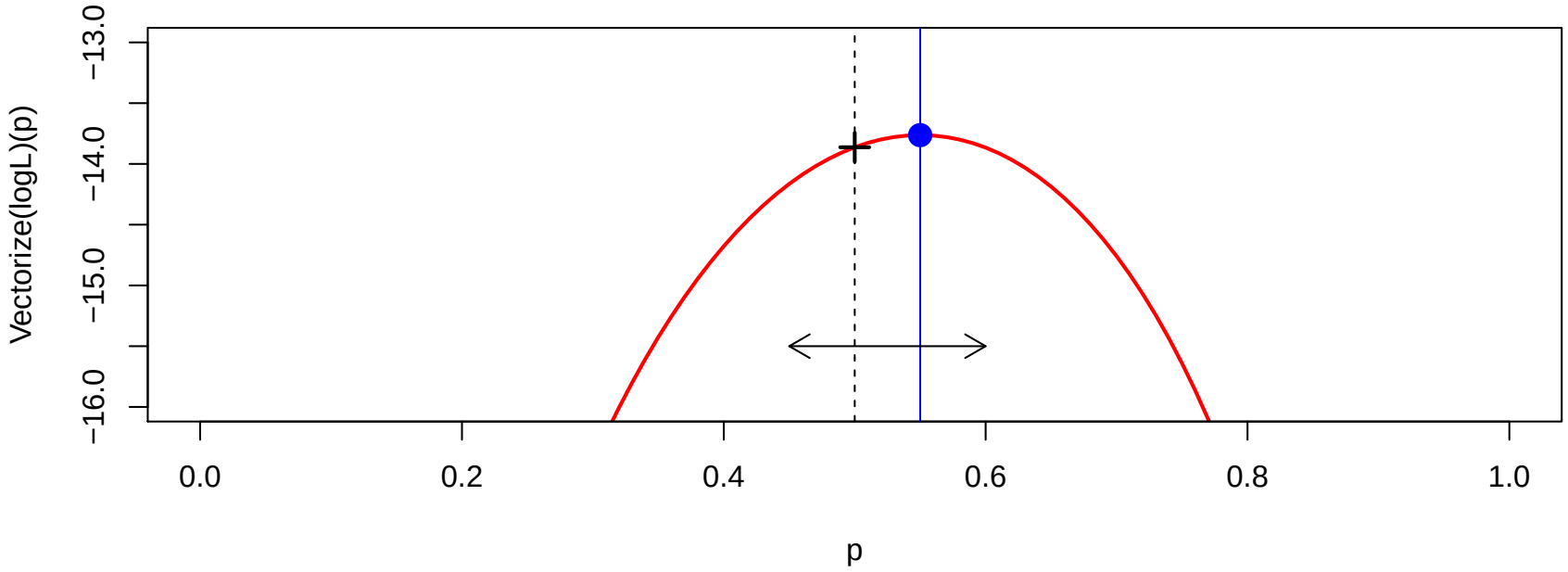
i.e. on retrouve ici $\widehat{\pi} = \overline{y}$.

Maintenant, on peut tester $H_0 : \pi = \pi_\star = 50\%$ versus $H_1 : \pi \neq 50\%$. Pour le test de Wald, on peut commencer par calculer $nI(\theta_\star)$ ou ,

```
>  nx=sum(X==1)
>  f = expression(nx*log(p)+(n-nx)*log(1-p))
>  Df = D(f, "p")
>  Df2 = D(Df, "p")
>  p=p0=0.5
>  (IF=-eval(Df2))
[1] 80
```

On peut d'ailleurs comparer cette valeur avec la valeur théorique, car $I(\pi)^{-1} = \pi(1-\pi)$

```
>  1/(p0*(1-p0)/n)
[1] 80
```



La statistique du test de Wald est ici

```
>  pml=optim(fn=neglogL,par=p0,method="BFGS")$par
>  (T=(pml-p0)^2*IF)
[1] 0.199997
```

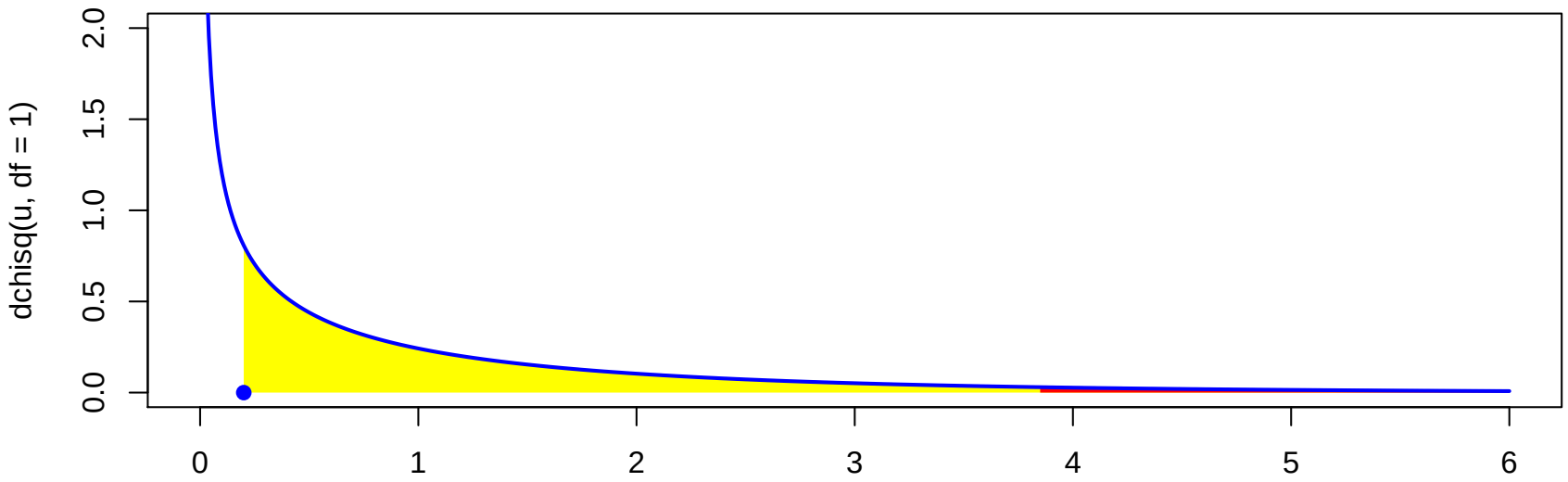
que l'on va chercher à comparer avec le quantile de la loi du χ^2,

```
>  T<qchisq(1-alpha,df=1)
[1] TRUE
```

i.e. on est dans la région d'acceptation du test. De manière duale, on peut calculer la p-value du test,

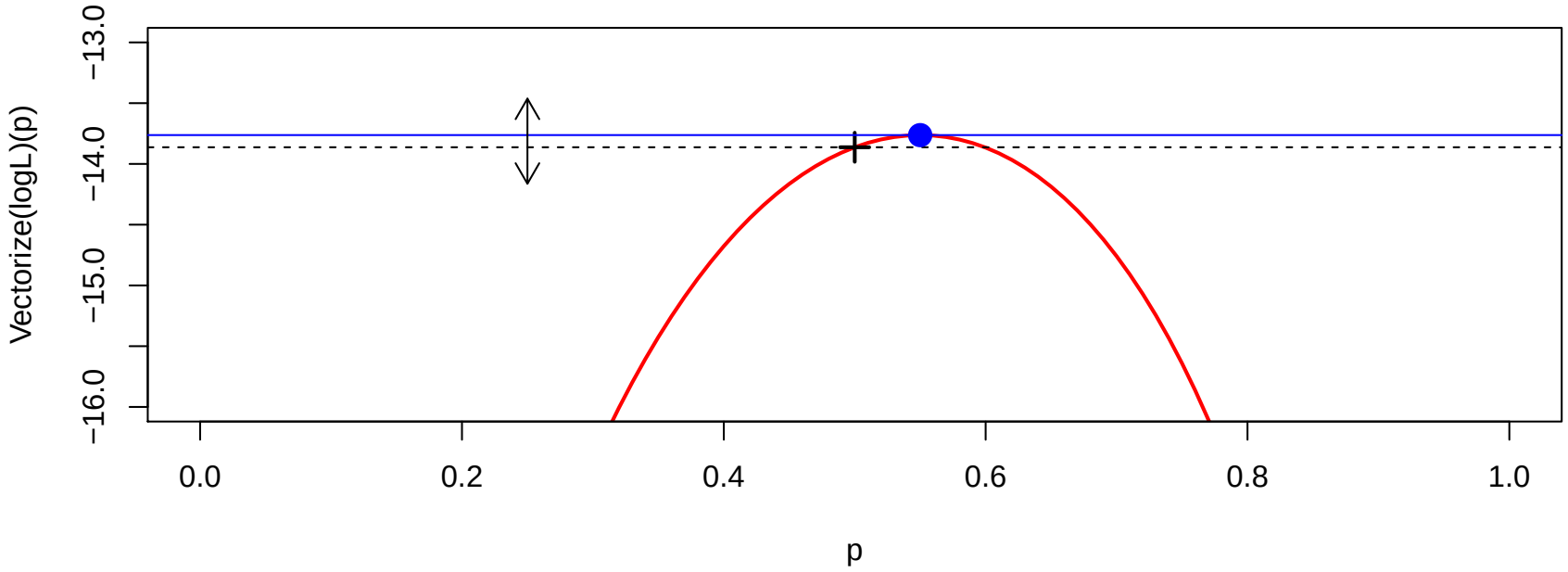
```
> 1-pchisq(T,df=1)
[1] 0.6547233
```

Donc on va accepter H_0.



Pour le test du rapport de vraisemblance, T est ici

```
>  (T=2*(logL(pml)-logL(p0)))
[1] 0.2003347
```



Là encore, on est dans la région d'acceptation,

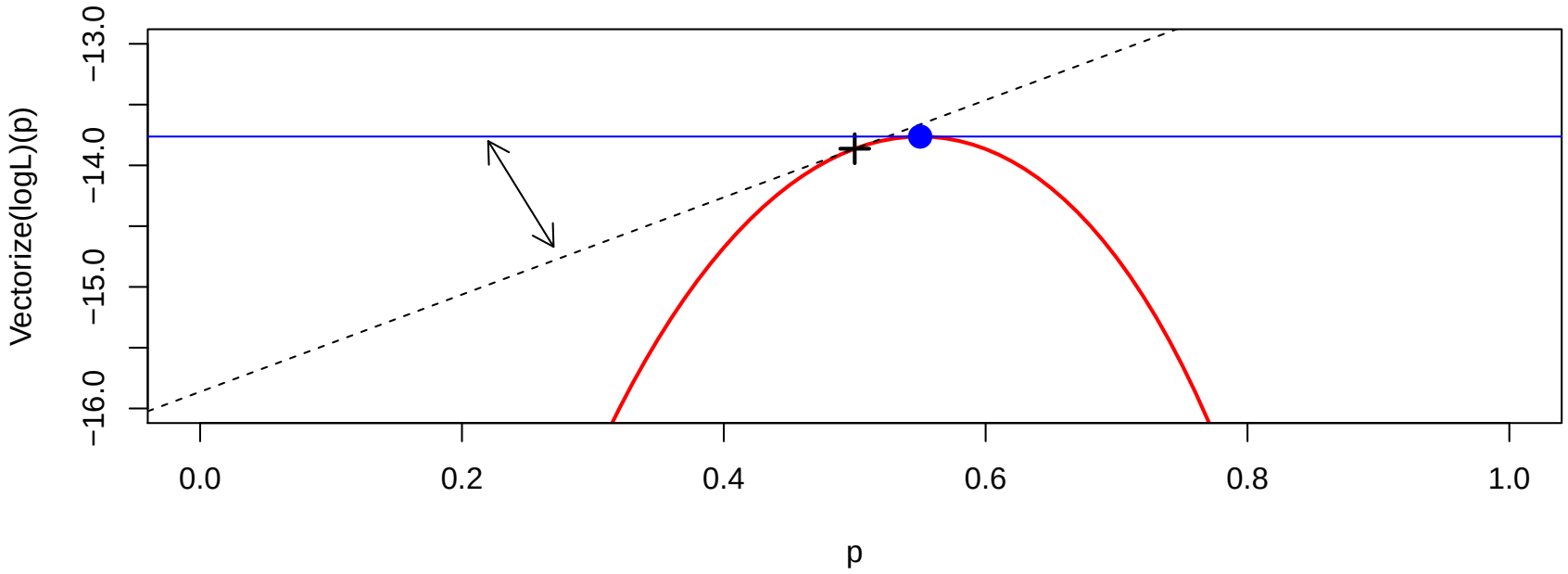
```
>  T<qchisq(1-alpha,df=1)
[1] TRUE
```

Enfin, on peut faire le test du score. Le score se calcule facilement,

```
> nx=sum(X==1)
> f = expression(nx*log(p)+(n-nx)*log(1-p))
> Df = D(f, "p")
> p=p0
> score=eval(Df)
```

La statistique de test est alors

```
>  (T=score^2/IF)
[1] 0.2
```



Là encore, on est dans la région d'acceptation,

```
>  T<qchisq(1-alpha,df=1)
[1] TRUE
```

Prise en compte de l'hétérogénéité

Supposons que π_i soit fonction de variables explicatives $\boldsymbol{X}_i$, e.g.

$$\pi_i = \mathbb{E}(Y_i|\boldsymbol{X}_i).$$

Le modèle classique est le modèle linéaire, $\mathbb{E}(Y_i|\boldsymbol{X}_i) = \boldsymbol{X}_i'\boldsymbol{\beta}$

On peut faire une estimation par moindres carrés pour estimer $\boldsymbol{\beta}$,

$$\widehat{\boldsymbol{\beta}} = \text{argmin}\left\{\sum_{i=1}^{n}[Y_i - \boldsymbol{X}_i'\boldsymbol{\beta}]^2, \boldsymbol{\beta} \in \mathbb{R}^k\right\}$$

Problème : $\pi_i \in [0,1]$ et $\boldsymbol{X}_i'\boldsymbol{\beta} \in \mathbb{R}$.

Problème : modèle non homoscédastique, en effet

$$\varepsilon_i = \begin{cases} 1 - \boldsymbol{X}_i'\boldsymbol{\beta} \text{ avec probabilité } \pi_i \\ -\boldsymbol{X}_i'\boldsymbol{\beta} \text{ avec probabilité } 1 - \pi_i \end{cases}$$

i.e. $\text{Var}(\varepsilon_) = \boldsymbol{X}_i'\boldsymbol{\beta} \cdot [1 - x\boldsymbol{X}_i'\boldsymbol{\beta}]$.

Prise en compte de l'hétérogénéité

```
baseavocat=read.table("http://freakonometrics.free.fr/AutoBI.csv",header=TRUE,sep=",")
avocat=(baseavocat$ATTORNEY==1)
sexe=rep("H",length(avocat)); sexe[baseavocat$CLMSEX==2]="F"
marital=rep("M",length(avocat)); marital[baseavocat$MARITAL==2]="C"
marital[baseavocat$MARITAL==3]="V"; marital[baseavocat$MARITAL==4]="D"
ceinture=(baseavocat$SEATBELT==1);
age=baseavocat$CLMAGE
cout=baseavocat$LOSS
base=data.frame(avocat,sexe,marital,ceinture,age,cout)
```

La base ici ressemble à

```
> tail(base,3)
     avocat sexe marital ceinture age  cout
1338  FALSE    F       M     TRUE  39 0.099
1339   TRUE    F       C     TRUE  18 3.277
1340  FALSE    F       C     TRUE  30 0.688
```

Utilisation de la cote (odds)

$\pi_i \in [0,1]$ alors que $\boldsymbol{X}_i'\boldsymbol{\beta} \in \mathbb{R}$, donc on ne peut pas supposer

$$\pi_i = \mathbb{E}(Y_i|\boldsymbol{X}_i) = \boldsymbol{X}_i'\boldsymbol{\beta}$$

$$\text{odds}_i = \frac{\pi_i}{1-\pi} \in [0,\infty].$$

ou

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

On appelle logit cette transormation,

$$\text{logit}(\pi_i) = \log\left(\frac{\pi_i}{1-\pi}\right)$$

Utilisation de la cote (odds)

On va supposer que

$$\text{logit}(\pi_i) = \boldsymbol{X}_i'\boldsymbol{\beta}$$

ou

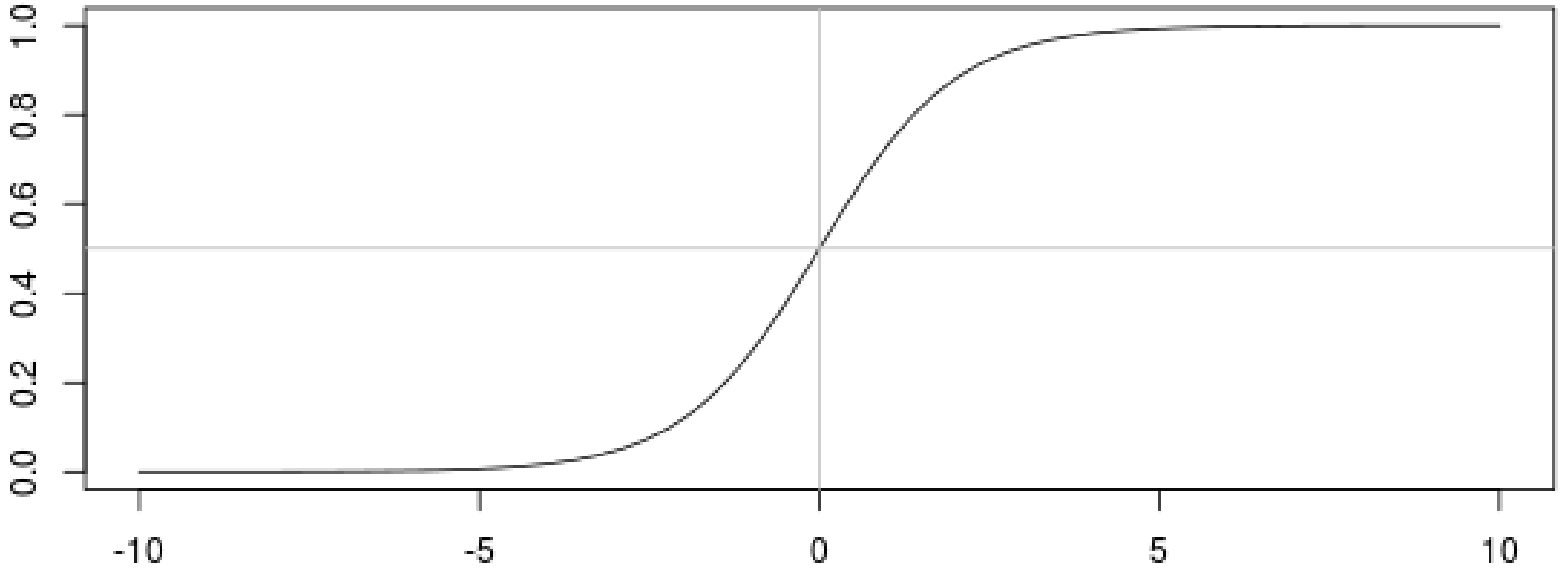
$$\pi_i = \text{logit}^{-1}(\boldsymbol{X}_i'\boldsymbol{\beta}) = \frac{\exp[\boldsymbol{X}_i'\boldsymbol{\beta}]}{1+\exp[\boldsymbol{X}_i'\boldsymbol{\beta}]}.$$

Remarque : si π_i est faible ($\pi_i \sim 0$) alors $\text{logit}^{-1} \sim \exp$, i.e.

$$\pi_i \sim \exp(\boldsymbol{X}_i'\boldsymbol{\beta})$$

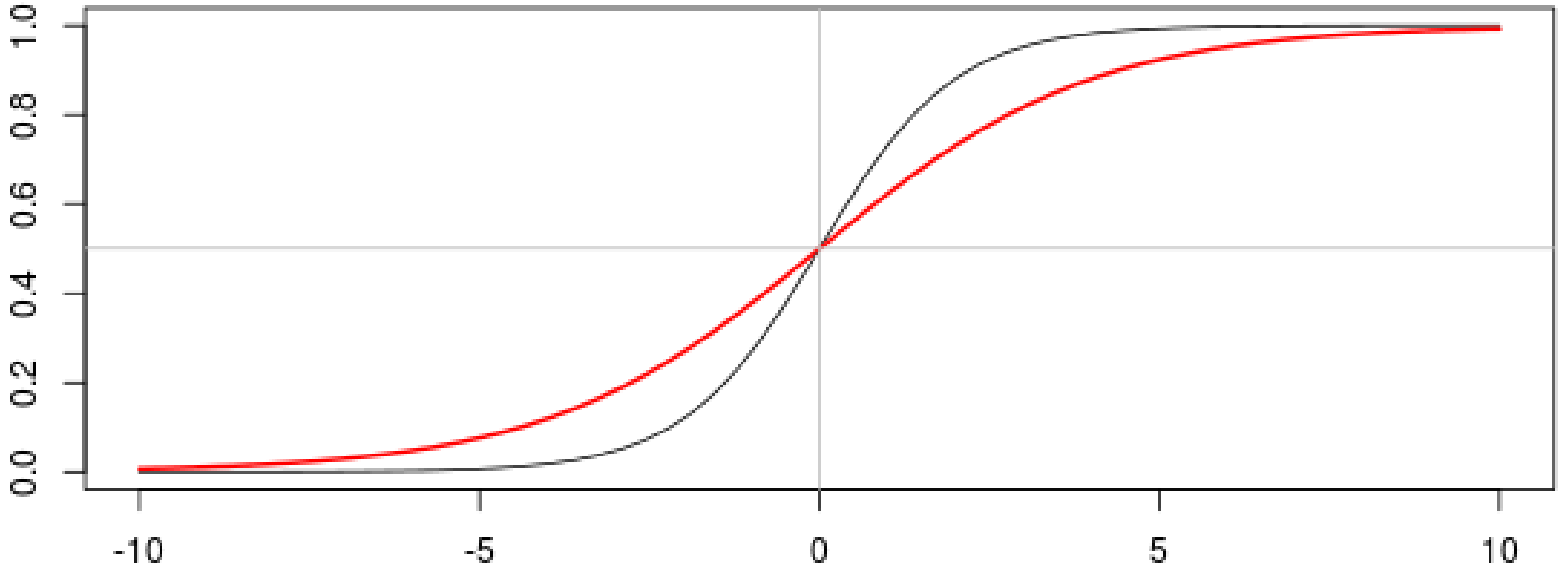
La fonction logistique

$$F(x) = \frac{\exp[x]}{1+\exp[x]}, \forall x \in \mathbb{R}.$$



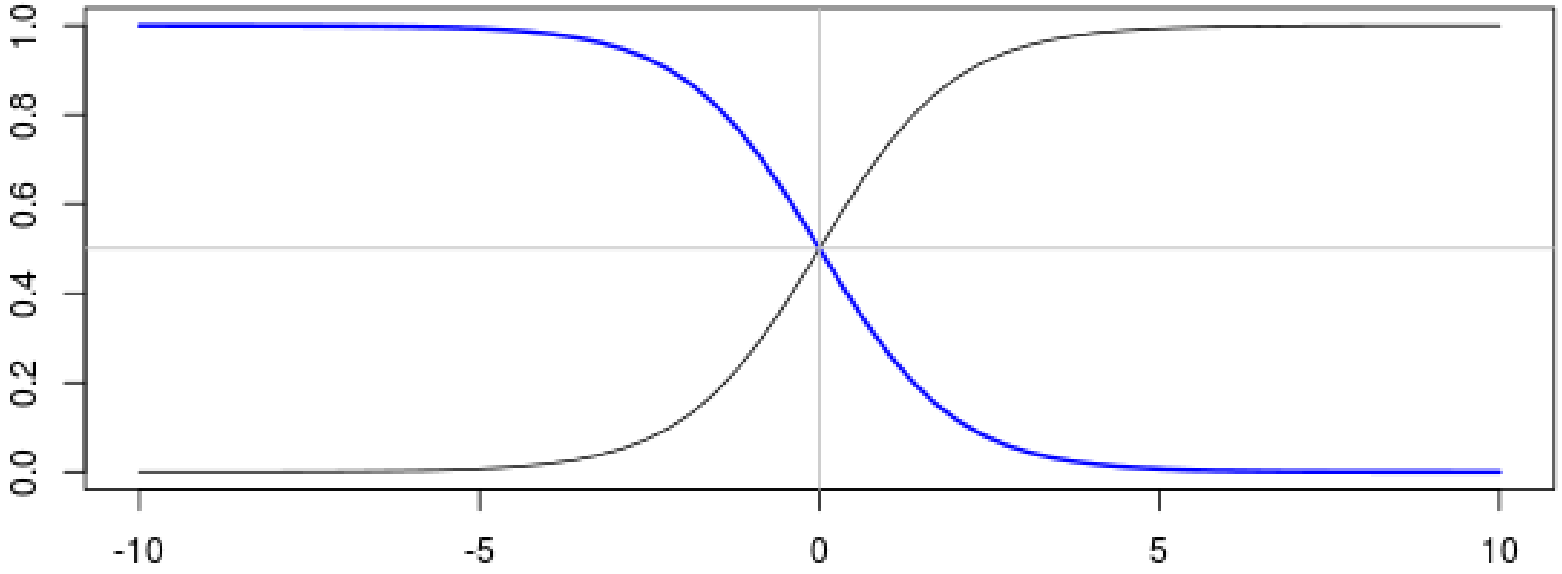
La fonction logistique

$$F(x) = \frac{\exp[2x]}{1+\exp[2x]}, \forall x \in \mathbb{R}.$$



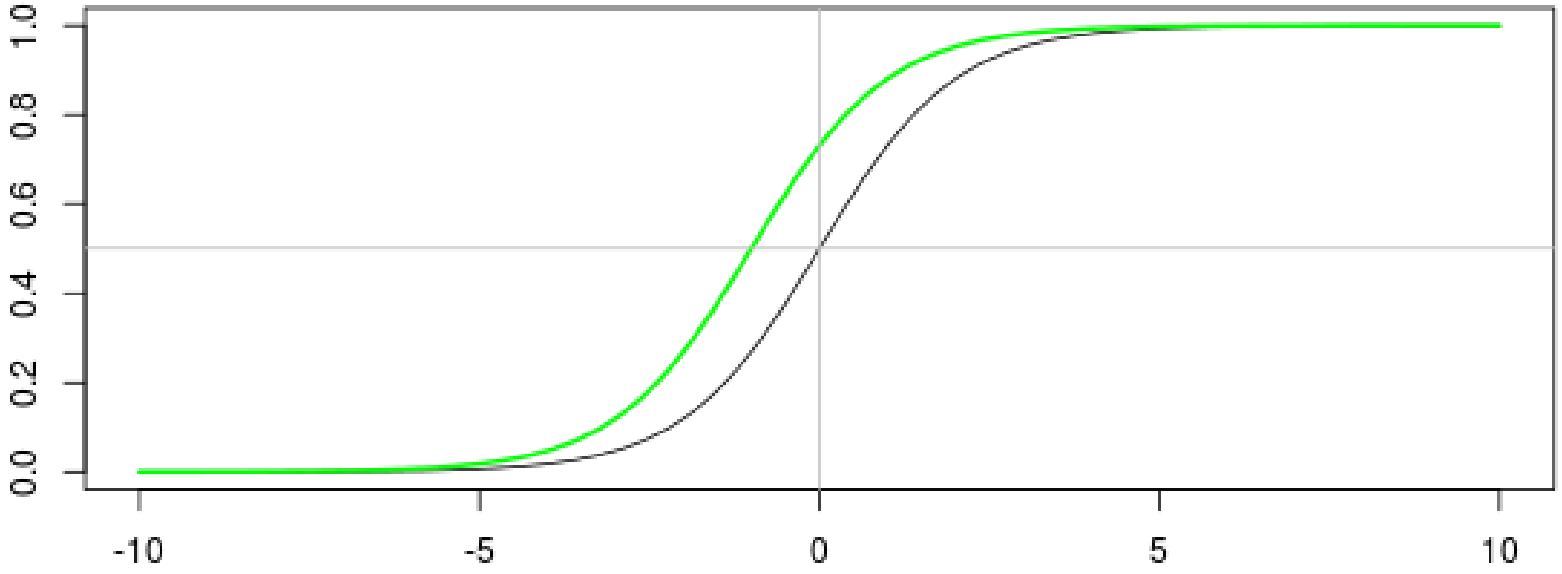
La fonction logistique

$$F(x) = \frac{\exp[-x]}{1+\exp[-x]}, \forall x \in \mathbb{R}.$$



La fonction logistique

$$F(x) = \frac{\exp[1+x]}{1+\exp[1+x]}, \forall x \in \mathbb{R}.$$



La régression logistique

La log-vraisemblance est ici

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

La résolution se fait en écrivant les conditions du premier ordre. Pour cela, on écrit le gradient $\nabla \log \mathcal{L}(\boldsymbol{\beta}) = [\partial \log \mathcal{L}(\boldsymbol{\beta})/\partial\beta_k]$, où

$$\frac{\partial \log \mathcal{L}(\boldsymbol{\beta})}{\partial \beta_k} == \sum_{i=1}^{n} \frac{y_i}{\pi_i(\boldsymbol{\beta})} \frac{\partial \pi_i(\boldsymbol{\beta})}{\partial \beta_k} - \frac{1-y_i}{\pi_i(\boldsymbol{\beta})} \frac{\partial \pi_i(\boldsymbol{\beta})}{\partial \beta_k}$$

or compte tenu de la forme de $\pi_i(\boldsymbol{\beta})$,

$$\frac{\partial \pi_i(\boldsymbol{\beta})}{\partial \beta_k} = \pi_i(\boldsymbol{\beta})[1-\pi_i(\boldsymbol{\beta})]X_{k,i}$$

on obtient

$$\frac{\partial \log \mathcal{L}(\boldsymbol{\beta})}{\partial \beta_k} == \sum_{i=1}^{n} X_{k,i}[y_i - \pi_i(\boldsymbol{\beta})]$$

Newton Raphson - Descente de gradient

Remarque Pour rechercher le zéro d'une fonction (réelle), i.e. $f(x) = 0$: on se donne x_0, et on pose

$$x_k = x_{k-1} - \frac{f(x_{k-1})}{f'(x_{k-1})}$$

Remarque Pour rechercher le maximum d'un fonction dérivable, on recherche le zéro de la dérivée, i.e. $h'(x) = 0$: on se donne x_0, et on pose

$$x_k = x_{k-1} - \frac{f'(x_{k-1})}{f''(x_{k-1})}$$

Il faut trouver numériquement le maximum de $\log \mathcal{L}(\boldsymbol{\beta})$.

L'algorithme standard est

1. partir d'une valeur initiale $\boldsymbol{\beta}_0$
2. poser $\boldsymbol{\beta}_k = \boldsymbol{\beta}_{k-1} - H(\boldsymbol{\beta}_{k-1})^{-1} \nabla \log \mathcal{L}(\boldsymbol{\beta}_{k-1})$

où $\nabla \log \mathcal{L}(\boldsymbol{\beta})$ est le gradient, et $H(\boldsymbol{\beta})$ la matrice Hessienne (on parle parfois de Score de Fisher).

Ici, le terme générique de la matrice Hessienne est

$$\frac{\partial^2 \log \mathcal{L}(\boldsymbol{\beta})}{\partial \beta_k \partial \beta_\ell} = \sum_{i=1}^{n} X_{k,i} X_{\ell,i} [y_i - \pi_i(\boldsymbol{\beta})]$$

Posons

$$\boldsymbol{\Omega} = [\omega_{i,j}] = \text{diag}(\widehat{\pi}_i(1 - \widehat{\pi}_i))$$

alors le gradient est

$$\nabla \log \mathcal{L}(\boldsymbol{\beta}) = \frac{\partial \log \mathcal{L}(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \boldsymbol{X}'(\boldsymbol{y} - \boldsymbol{\pi})$$

alors que la matrice Hessienne estimation

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

Aussi, l'algorithme de Newton Raphson s'écrit

$$\boldsymbol{\beta}_k = (\boldsymbol{X}'\boldsymbol{\Omega}\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{\Omega}\boldsymbol{Z} \text{ où } \boldsymbol{Z} = \boldsymbol{X}\boldsymbol{\beta}_{k-1} + \boldsymbol{X}'\boldsymbol{\Omega}^{-1}\boldsymbol{y} - \boldsymbol{\pi}),$$

On reconnait une régression pondérée itérée,

$$\boldsymbol{\beta}_k = \text{argmin}\,\{(\boldsymbol{Z} - \boldsymbol{X}\boldsymbol{\beta})'\boldsymbol{\Omega}(\boldsymbol{Z} - \boldsymbol{X}\boldsymbol{\beta}).\}$$

D'un point de vue computationnel, l'estimation se fait de la manière suivante :

```
X=cbind(rep(1,nrow(base)),base$cout)
Y=base$avocat*1
```

on part d'une valeur initiale (e.g. une estimation classique de modèle linéaire)

```
beta=lm(Y~0+X)$coefficients
```

On fait ensuite une boucle,

```
for(s in 1:20){
pi=exp(X%*%beta)/(1+exp(X%*%beta))
gradient=t(X)%*%(Y-pi)
omega=matrix(0,nrow(X),nrow(X));diag(omega)=(pi*(1-pi))
hesienne=-t(X)%*%omega%*%X
beta=beta-solve(hesienne)%*%gradient}
```

Ici, on obtient les valeurs suivantes

```
             [,1]        [,2]
 [1,]  0.50034948 0.001821554
 [2,] -0.05689403 0.015434935
 [3,] -0.19941087 0.055570172
 [4,] -0.45629828 0.140490153
 [5,] -0.73627462 0.253154397
 [6,] -0.85830164 0.309712845
 [7,] -0.87192186 0.316615292
 [8,] -0.87207800 0.316697581
```

i.e. on converge (et vite).

Comportement asymptotique (et intervalle de confiance)

On peut montrer que $\widehat{\boldsymbol{\beta}} \xrightarrow{\mathbb{P}} \boldsymbol{\beta}$ et

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

Numériquement, la encore, on peut approcher $I(\boldsymbol{\beta})^{-1}$ qui est la variance (asymptotique) de notre estimateur. Or $I(\boldsymbol{\beta}) = H(\boldsymbol{\beta})$, donc les écart-types de $\widehat{\boldsymbol{\beta}}$ sont

```
> pi=exp(X%*%beta)/(1+exp(X%*%beta))
+ omega=matrix(0,nrow(X),nrow(X));diag(omega)=(pi*(1-pi))
+ hesienne=-t(X)%*%omega%*%X
+ sqrt(-diag(solve(hesienne)))
[1] 0.09128379 0.02902411
```

On retrouve toutes ces valeurs en utilisant

```
> reglogit=logit(Y~0+X)

Coefficients:
   Estimate Std. Error z value Pr(>|z|)
X1 -0.87208    0.09128  -9.553   <2e-16 ***
X2  0.31670    0.02902  10.912   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1857.6  on 1340  degrees of freedom
Residual deviance: 1609.4  on 1338  degrees of freedom
AIC: 1613.4

Number of Fisher Scoring iterations: 8

Message d’avis :
glm.fit: fitted probabilities numerically 0 or 1 occurred
```

Test d'hypothèse et significativité

Considérons un test $H_0 : \beta_k = 0$ contre $H_1 : \beta_k \neq 0$ (classique dans le modèle linéaire).

Compte tenu du comportement asymptotiquement Gaussien des estimateurs, on a automatiquement un test de significativité des coefficients : sous H_0,

$$\frac{\widehat{\beta}_k}{\sqrt{\widehat{\text{Var}}(\widehat{\beta}_k)}} \sim \mathcal{N}(0,1).$$

Remarque : ce test correspond à un test de Wald (d'où la notation z sous R)

Régression linéaire sur une variable catégorielle

Supposons que l'on régresse sur X, une variable binaire, i.e $X \in \{0, 1\}$, avec

$$\mathbb{P}(Y = 1|X) = \beta_0 + \beta_1 X$$

qui prend les valeurs β_0 si $x = 0$ (modalité de référence) et $\beta_0 + \beta_1$ si $x = 1$.

La log-vraisemblance devient

$$\begin{aligned}\log \mathcal{L}(\beta_0, \beta_1) &= \sum_{i, x_i=0} y_i \log[\beta_0] + \sum_{i, x_i=1} y_i \log[\beta_0 + \beta_1] \\ &\quad + \sum_{i, x_i=0} (1 - y_i) \log[1 - \beta_0] + \sum_{i, x_i=1} (1 - y_i) \log[1 - \beta_0 + \beta_1]\end{aligned}$$

et les conditions du premier ordre donnent

$$\widehat{\beta_0} = \frac{\sum_{x_i=0} y_i}{\sum_{x_i=0} 1} \text{ et } \widehat{\beta_1} = \frac{\sum_{x_i=1} y_i}{\sum_{x_i=0} 1} - \frac{\sum_{x_i=0} y_i}{\sum_{x_i=0} 1}$$

Régression linéaire sur une variable catégorielle

Sur nos données, on peut considérer le sexe comme variable explicative.

```
> xtabs(~avocat + sexe, data=base)
       sexe
avocat    F   H
  FALSE 390 265
  TRUE  352 333
```

Si on fait la régression linéaire

```
> reglm = lm(avocat ~ sexe, data=base)
> summary(reglm)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.47439    0.01830   25.92  < 2e-16 ***
sexeH        0.08246    0.02740    3.01  0.00266 **
```

Régression linéaire sur une variable catégorielle

Ces paramètres correspondent aux fréquences empiriques,

```
> attach(base)
> sum((avocat==1)&(sexe=="F"))/sum((sexe=="F"))
[1] 0.4743935
> sum((avocat==1)&(sexe=="H"))/sum((sexe=="H"))-
+ sum((avocat==1)&(sexe=="F"))/sum((sexe=="F"))
[1] 0.08246266
```

Régression logistique sur une variable catégorielle

Avec un modèle logistique, on obtient les mêmes prédictions,

```
> reglogit = logit(avocat ~ sexe, data=base)
> summary(reglogit)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.10252    0.07352  -1.394  0.16319
sexeH        0.33093    0.11037   2.998  0.00271 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> exp(reglogit$coefficients[1])/
+ (1+exp(reglogit$coefficients[1]))
(Intercept)
  0.4743935
> exp(sum(reglogit$coefficients[1:2]))/
+ (1+exp(sum(reglogit$coefficients[1:2])))
[1] 0.5568562
```

Modèle alternatif, le modèle probit

Soit H une fonction de répartition d'une variable rélle, $H : \mathbb{R} \mapsto [0,1]$. On suppose ici

$$\pi_i = H(\boldsymbol{X}_i'\boldsymbol{\beta}) \text{ ou } \boldsymbol{X}_i'\boldsymbol{\beta} = H^{-1}(\pi_i)$$

- si H est la fonction de répartition de la loi logistique, on retrouve le modèle logit,
- si H est la fonction de répartition de la loi $\mathcal{N}(0,1)$, on obtient le modèle probit,

Remarque : probit signifie probability unit.

On suppose qu'il existe une variable latente $Y_i^\star$ non observée, telle que

$$\pi_i = \mathbb{P}(Y_i = 1) = \mathbb{P}(Y_i^\star > 0)$$

On suppose que $Y_i^\star = \boldsymbol{X}_i'\boldsymbol{\beta} + \epsilon_i$, où ϵ_i est une erreur de loi H. Alors

$$\pi_i = \mathbb{P}(Y_i = 1) = \mathbb{P}(Y_i^\star > 0) = \mathbb{P}(\epsilon_i > -\boldsymbol{X}_i'\boldsymbol{\beta}) = 1 - H(-\boldsymbol{X}_i'\boldsymbol{\beta})$$

alors que

$$1-\pi_i = \mathbb{P}(Y_i = 0) = \mathbb{P}(Y_i^\star \leq 0) = \mathbb{P}(\epsilon_i \leq -\boldsymbol{X}_i'\boldsymbol{\beta}) = H(-\boldsymbol{X}_i'\boldsymbol{\beta})$$

Si $H = \Phi$, alors H est symmétrique par rapport à 0, et

$$\pi_i = \mathbb{P}(Y_i = 1) = 1 - \Phi(-\boldsymbol{X}_i'\boldsymbol{\beta}) = \Phi(\boldsymbol{X}_i'\boldsymbol{\beta}),$$

ou encore $\eta_i = \boldsymbol{X}_i'\boldsymbol{\beta} = \Phi^{-1}(\pi_i)$.

Si $H(x) = \dfrac{e^x}{1+e^x}$, on a une loi logistique, qui est aussi symmétrique par rapport à 0, et

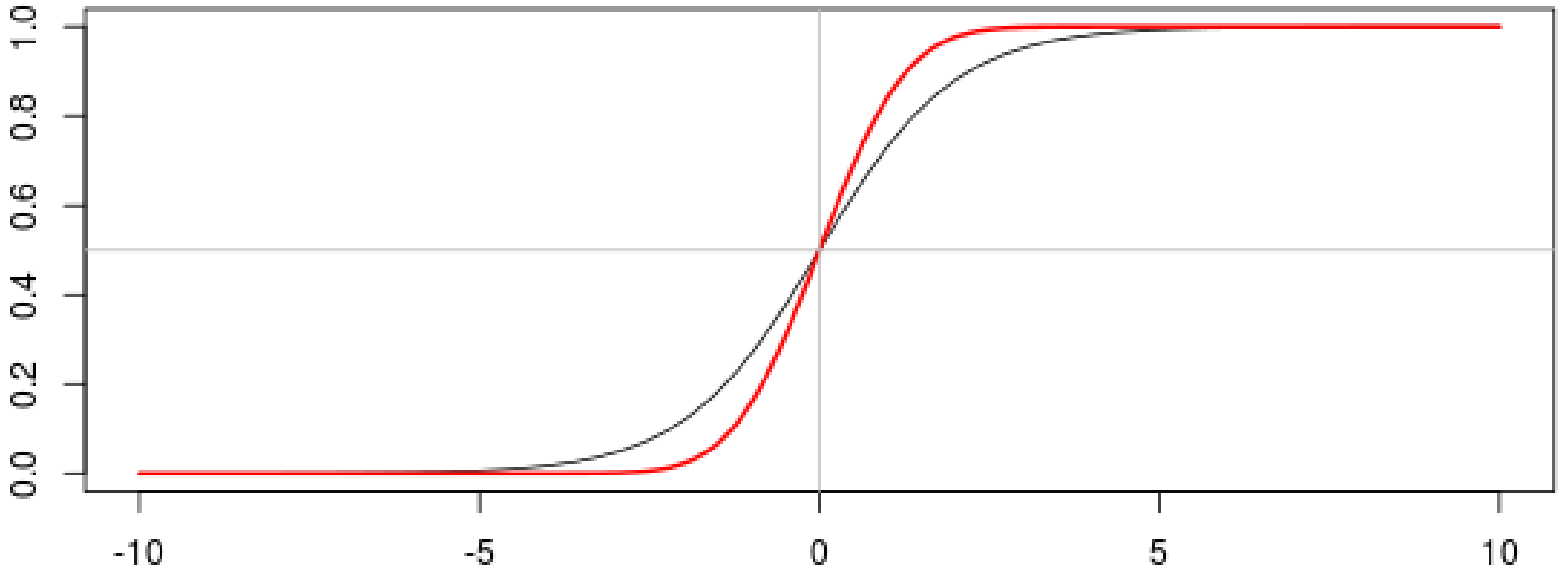
$$\pi_i = \mathbb{P}(Y_i = 1) = 1 - H(-\boldsymbol{X}_i'\boldsymbol{\beta}) = H(\boldsymbol{X}_i'\boldsymbol{\beta}),$$

ou encore $\eta_i = \boldsymbol{X}_i'\boldsymbol{\beta} = H^{-1}(\pi_i)$.

Remarque : pourquoi prendre un seuil nul ? sinon le modèle ne serait pas identificable. De même si on ne fixe pas la variance des résidus. On prend donc (arbitrairement) $s = 0$ et $\sigma^2 = 1$.

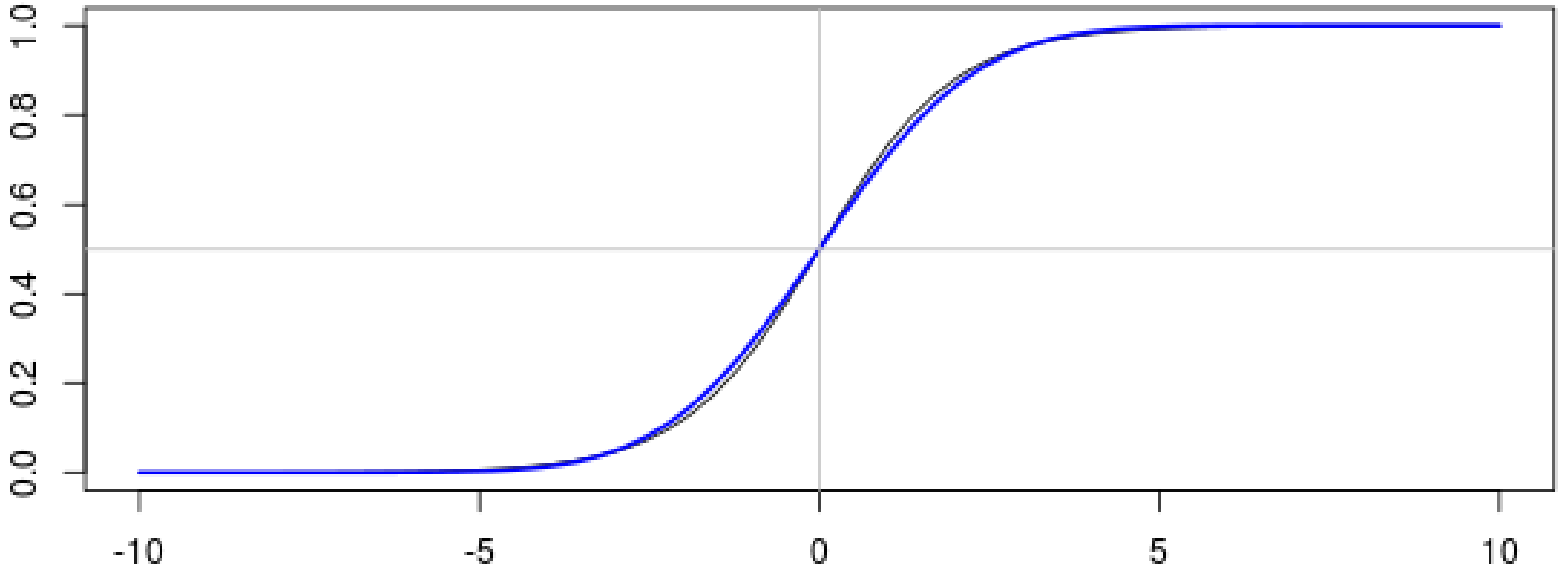
La fonction logistique

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left(-\frac{z^2}{2}\right) dz, \forall x \in \mathbb{R}.$$



La fonction logistique

$$\Phi(x) = \frac{\sqrt{3}}{\pi\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left(-\frac{3z^2}{2\pi^2}\right) dz, \forall x \in \mathbb{R}.$$



Qualité de l'ajustement

Pour juger de la validité du modèle, on peut considérer plusieurs quantités.

On peut définir la déviance, qui compare la log-vraisemblance du modèle et cellle du modèle parfait,

$$D = -2\sum_{i=1}^{n}[Y_i \log[\widehat{\pi}_i] + (1-Y_i)\log[1-\widehat{\pi}_i]]$$

Plus la variance est faible, meilleur est le modèle .

On peut aussi considérer la statistique de Pearson

$$\chi^2 = \sum_{i=1}^{n} \frac{(y_i - \widehat{\pi}_i)^2}{\widehat{\pi}_i[1-\widehat{\pi}_i]}$$

Étude des résidus

A l'aide de la relation précédante, on peut définir

$$\widehat{\varepsilon}_i = \frac{Y_i - \widehat{\pi}_i}{\sqrt{\widehat{\pi}_i[1-\widehat{\pi}_i]}}$$

On parlera de résidus de Pearson.

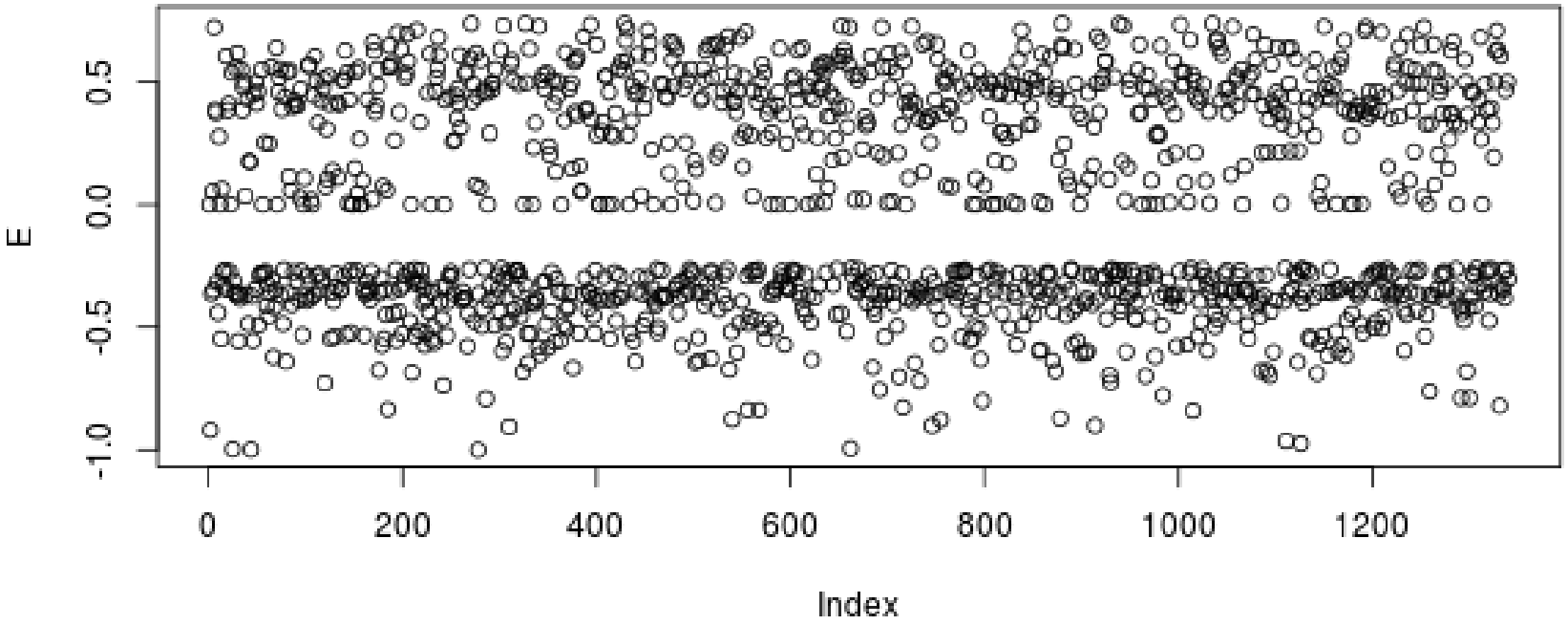
Il existe aussi les résidus de déviance.

$$\widehat{\varepsilon}_i = \pm[Y_i \log[\widehat{\pi}_i] + (1-Y_i)\log[1-\widehat{\pi}_i]]^{\frac{1}{2}}$$

avec un signe positif si $Y_i \geq \widehat{\pi}_i$, négatif sinon.

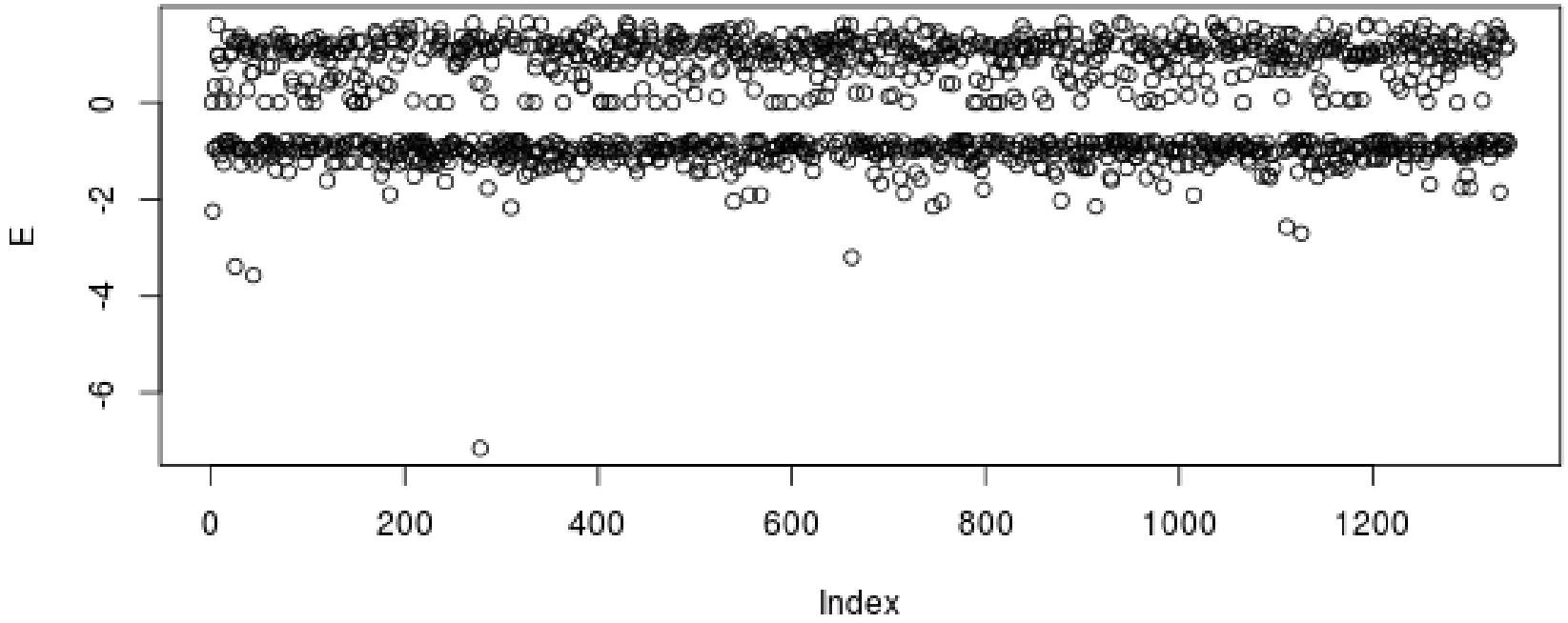
Visualisation des résidus

```
> reglogit = logit(avocat ~ cout+sexe, data=base)
> E = residuals(reglogit,type=‘‘response’’)
```



Visualisation des résidus

```
> reglogit = logit(avocat ~ cout+sexe, data=base)
> E = residuals(reglogit,type=‘‘deviance’’)
```



Prédiction avec un modèle logistique

Sous R, la prédiction sera (par défaut) une prédiction du score, i.e. $\boldsymbol{x}'\widehat{\boldsymbol{\beta}}$.

```
> predict(reglogit,newdata=data.frame(cout=10,sexe="H"))
       1
2.530417
```

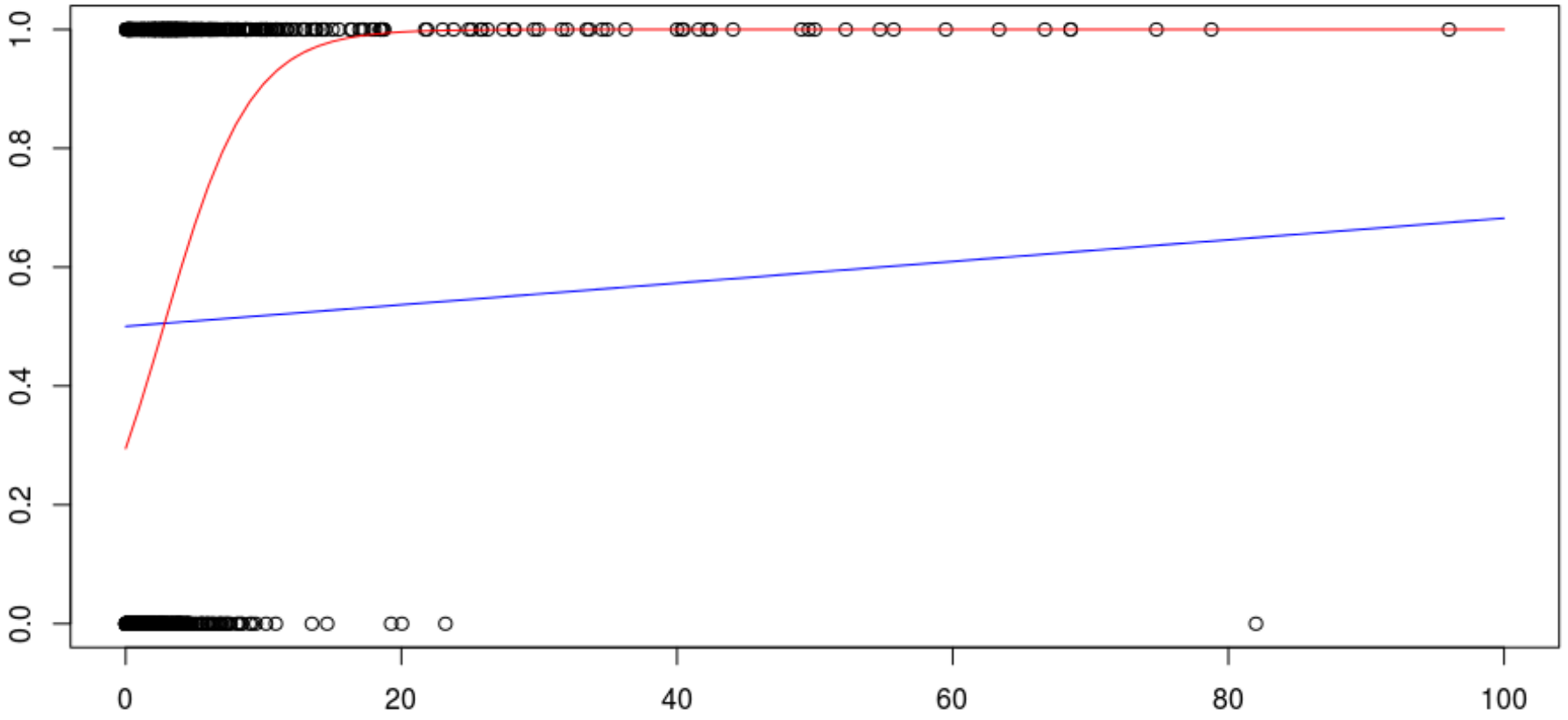
Il faut alors utiliser la loi logistique pour prévoir π,

```
> exp(predict(reglogit,newdata=data.frame(cout=10,sexe="H")))/
+ (1+exp(predict(reglogit,newdata=data.frame(cout=10,sexe="H"))))
        1
0.9262468
```

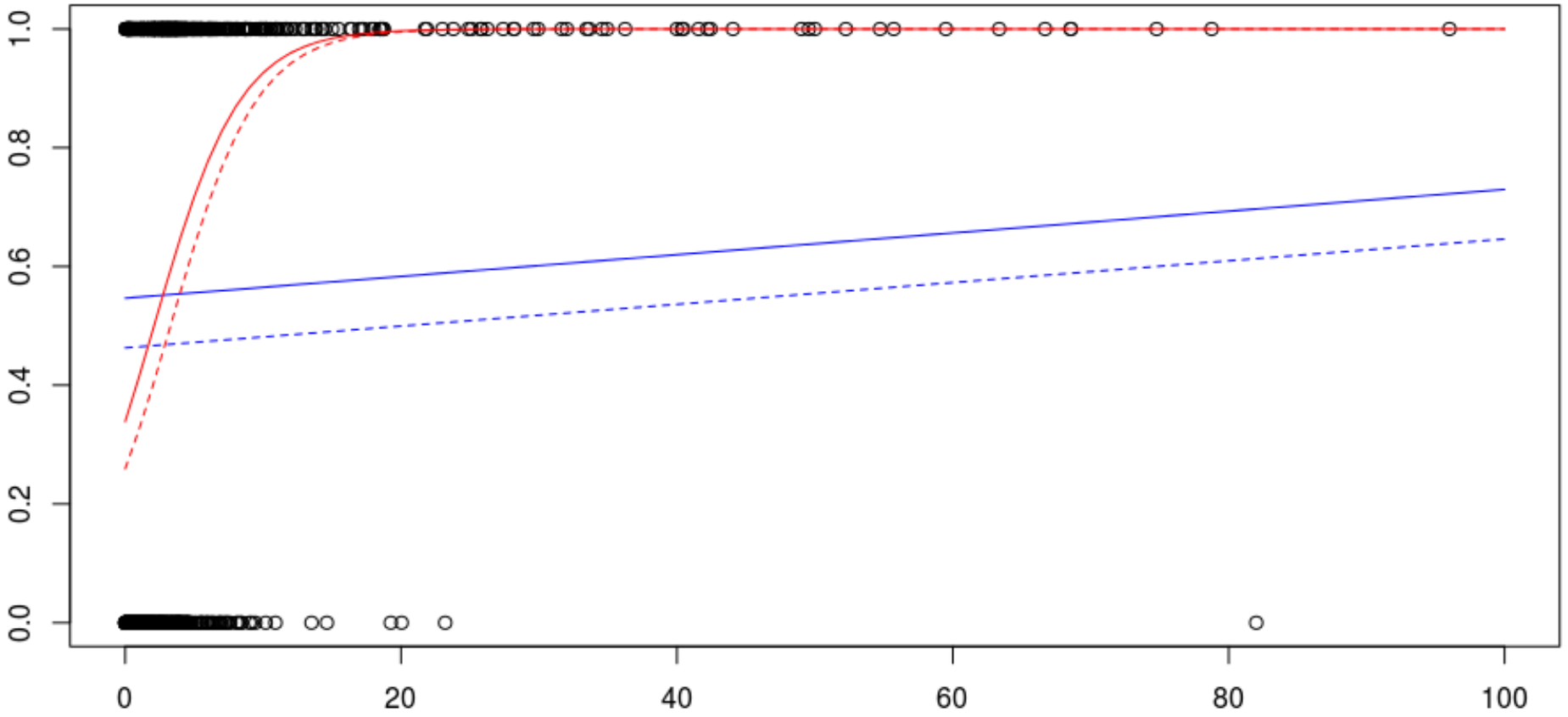
ou utiliser directement

```
> predict(reglogit,newdata=data.frame(cout=10,sexe="H"),type="response")
        1
0.9262468
```

Prédiction avec un modèle logistique



Prédiction avec un modèle logistique



Pour juger de la qualité du modèle, plusieurs outils existent.

Courbe ROC, receiver operating characteristic

On dispose d'observations Y_i prenant les valeurs 0 (on dira **négatives**) ou 1 (on dira **positives**).

On a construit un modèle qui prédit π_i. En se fixant un seuil s, si $\widehat{\pi}_i \leq s$, on prédit $\widehat{Y}_i = 0$, et si $\widehat{\pi}_i > s$, on prédit $\widehat{Y}_i = 1$.

Le modèle sera bon si les positifs sont prédits positifs, et les négatifs sont prédits négatifs.

Le choix du seuil s permettra de minimiser soit les faux positifs, soit les faux négatifs.

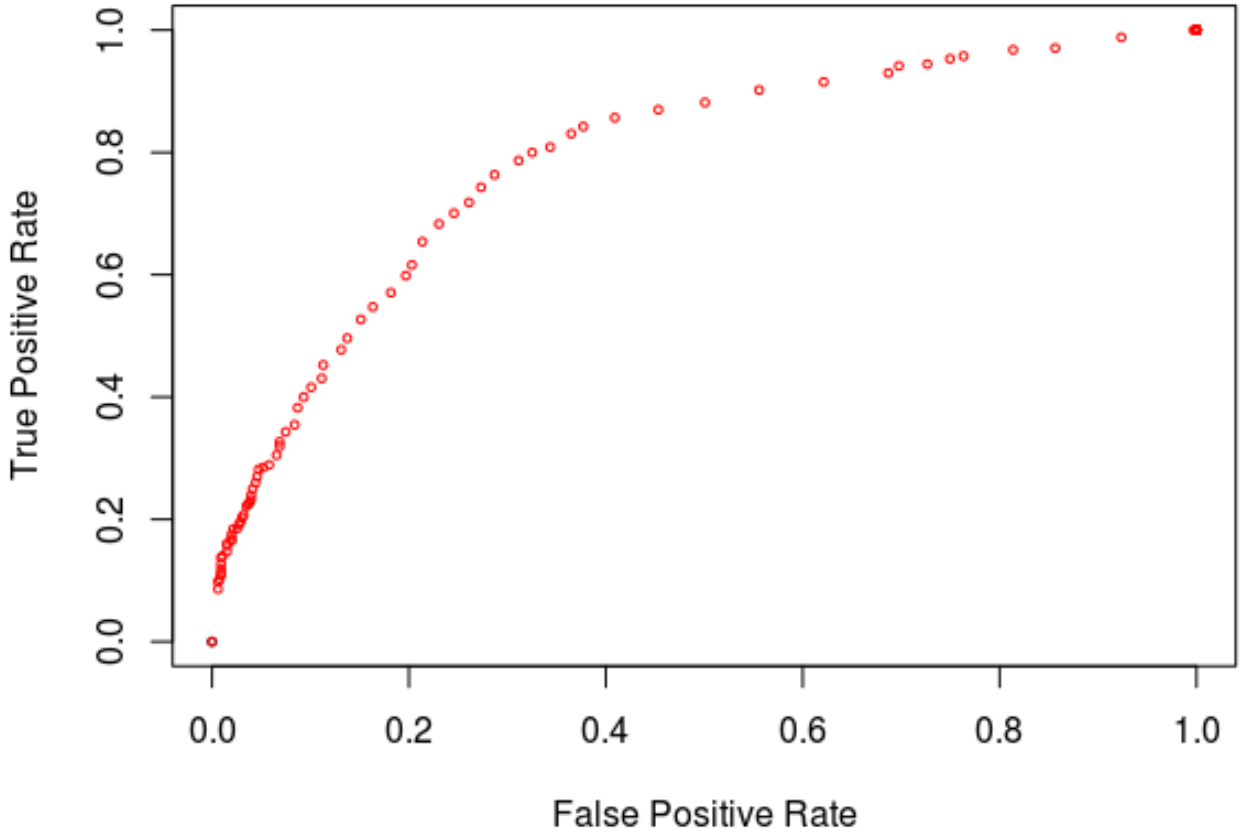
	$Y = 1$	$Y = 0$
$\widehat{Y}_i = 1$	TP	FP
$\widehat{Y}_i = 0$	FN	TN

où TP sont les True Positive, FP les False Positive, etc.

Courbe ROC, receiver operating characteristic

```
S=predict(reglogit,type="response")
plot(0:1,0:1,xlab="False Positive Rate",
ylab="True Positive Rate",cex=.5)
for(s in seq(0,1,by=.01)){
Ps=(S>s)*1
FP=sum((Ps==1)*(avocat==0))/sum(avocat==0)
TP=sum((Ps==1)*(avocat==1))/sum(avocat==1)
points(FP,TP,cex=.5,col="red")}
```

Courbe ROC, receiver operating characteristic

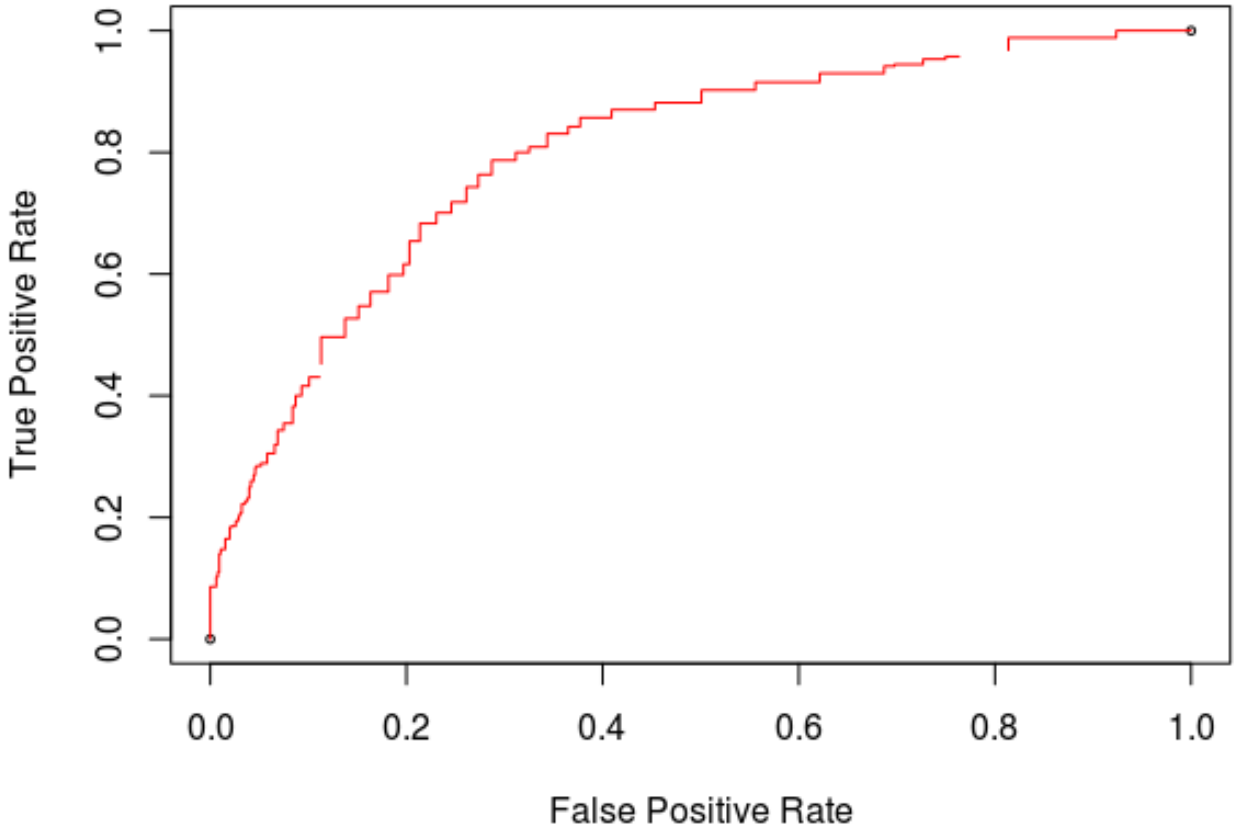


Courbe ROC, receiver operating characteristic

En reliant les points, on obtient la courbe ROC,

```
FP=TP=rep(NA,101)
plot(0:1,0:1,xlab="False Positive Rate",
ylab="True Positive Rate",cex=.5)
for(s in seq(0,1,by=.01)){
Ps=(S>s)*1
FP[1+s*100]=sum((Ps==1)*(avocat==0))/sum(avocat==0)
TP[1+s*100]=sum((Ps==1)*(avocat==1))/sum(avocat==1)
}
lines(c(FP),c(TP),type="s",col="red")
```

Courbe ROC, receiver operating characteristic

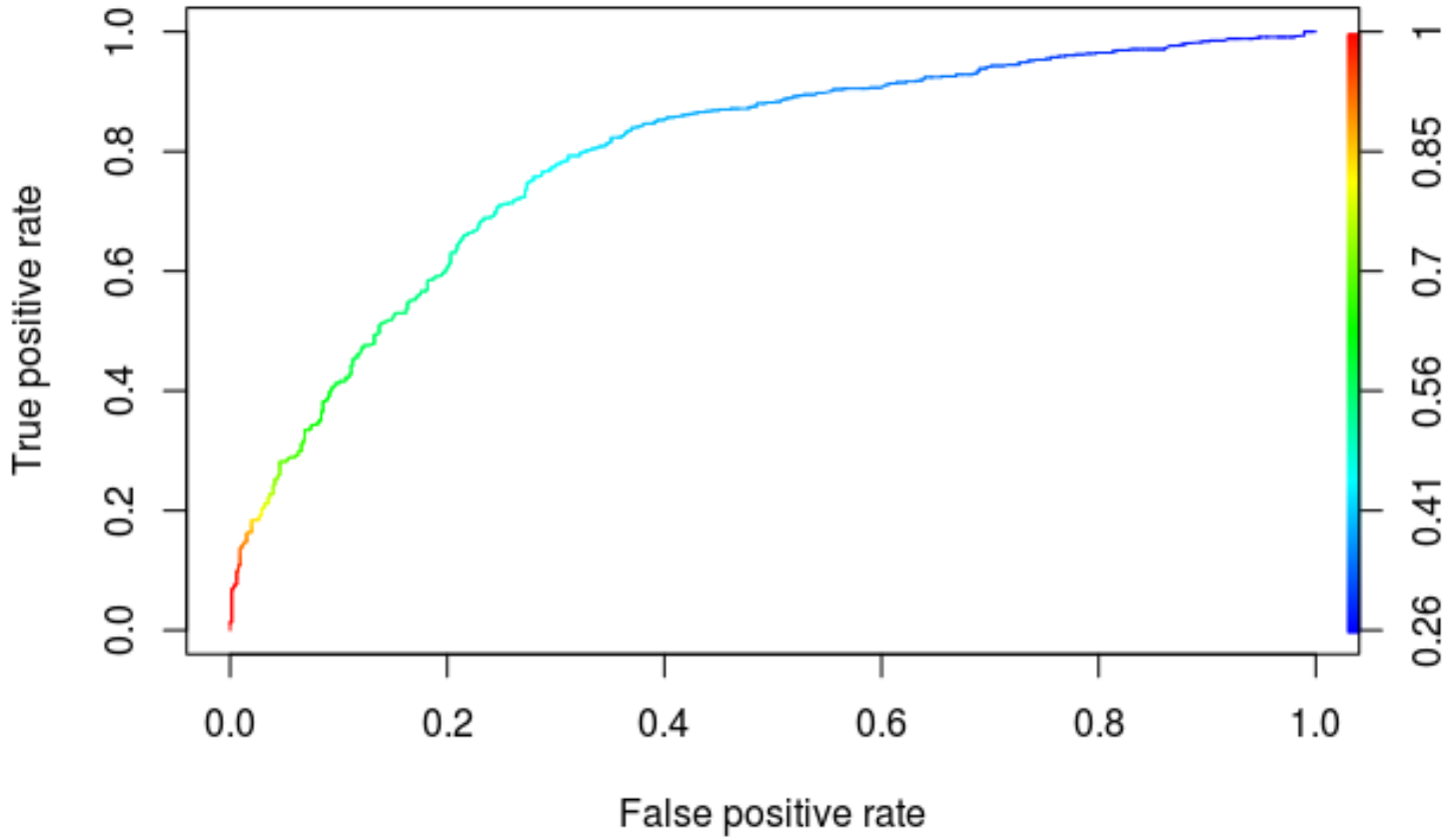


Courbe ROC, receiver operating characteristic

Mais il existe aussi des librairies de fonctions dédiées à cette analyse,

```
library(ROCR)
pred=prediction(S,avocat)
perf=performance(pred,"tpr", "fpr")
plot(perf,colorize = TRUE)
```

Courbe ROC, receiver operating characteristic



Arbre de régression

On va utiliser la méthode CART (Classification And Regression Trees), introduite par Breiman (1984). Pour aller plus loin, Berk (2008), chapitre 3.

Un arbre de décision est une règle de classification basé sur des tests associés aux attributs organisés de manière arborescente.

Considérons - pour commencer - une variable continue X_1 (ou X), et une variable à expliquer Y qui prend les valeurs $\{0, 1\}$.

Arbre de régression - indice de Gini

Étant donne une partition $\mathcal{S} = \{S_1, \cdots, S_k\}$,

$$\text{gini}(S) = \sum_{i=1}^{k} \frac{|S_i|}{|S|} \cdot \left(1 - \frac{S_i}{S}\right) = \sum_{i \neq j} \frac{|S_i|}{|S|} \cdot \frac{S_j}{S}$$

Dans une régression de Y sur X (partitionné en 2 classes, notées A et B),

$$\text{gini}(Y|X) = -\sum_{x \in \{A,B\}} \frac{n_x}{n} \sum_{y \in \{0,1\}} \frac{n_{x,y}}{n_x} \left(1 - \frac{n_{x,y}}{n_x}\right)$$

Dans une régression de Y sur X (partitionné en 2 classes, notées A et B),

$$\text{gini}(Y|X) = -\sum_{x \in \{A,B,C\}} \frac{n_x}{n} \sum_{y \in \{0,1\}} \frac{n_{x,y}}{n_x} \left(1 - \frac{n_{x,y}}{n_x}\right)$$

Arbre de régression - indice de Gini

Le code est ici

```
> u=sort(unique(B$age))
> gini=(NA,length(u))
> for(i in 2:(length(u))){
+ I=B$age<(u[i]-.5)
+ ft1=sum(B$Y[I==TRUE])/nrow(B)
+ ff1=sum(B$Y[I==FALSE])/nrow(B)
+ ft0=sum(1-B$Y[I==TRUE])/nrow(B)
+ ff0=sum(1-B$Y[I==FALSE])/nrow(B)
+ ft=ft0+ft1
+ f0=ft0+ff0
+ gini[i] = -((ft1+ft0)*(ft1/(ft1+ft0)*(1-ft1/(ft1+ft0))+
+           ft0/(ft1+ft0)*(1-ft0/(ft1+ft0))) +
+                  (ff1+ff0)*(ff1/(ff1+ff0)*(1-ff1/(ff1+ff0))+
+           ff0/(ff1+ff0)*(1-ff0/(ff1+ff0))))
+ }
```

Arbre de régression - entropie

$$\text{entropie}(S) = -\sum_{i=1}^{k} \frac{|S_i|}{|S|} \cdot \log\left(\frac{S_i}{S}\right)$$

Dans une régression de Y sur X (partitionné en 2 classes, notées A et B),

$$\text{entropie}(Y|X) = -\sum_{x\in\{A,B\}} \frac{n_x}{n} \sum_{y\in\{0,1\}} \frac{n_{x,y}}{n_x} \log\left(\frac{n_{x,y}}{n_x}\right)$$

Dans une régression de Y sur X (partitionné en 2 classes, notées A, B et C),

$$\text{entropie}(Y|X) = -\sum_{x\in\{A,B,C\}} \frac{n_x}{n} \sum_{y\in\{0,1\}} \frac{n_{x,y}}{n_x} \log\left(\frac{n_{x,y}}{n_x}\right)$$

Arbre de régression - entropie

```
> u=sort(unique(B$age))
> entropie=rep(NA,length(u))
> for(i in 2:(length(u))){
+ I=B$age<(u[i]-.5)
+ ft1=sum(B$Y[I==TRUE])/nrow(B)
+ ff1=sum(B$Y[I==FALSE])/nrow(B)
+ ft0=sum(1-B$Y[I==TRUE])/nrow(B)
+ ff0=sum(1-B$Y[I==FALSE])/nrow(B)
+ ft=ft0+ft1
+ f0=ft0+ff0
+ entropie[i] = -((ft1+ft0)*(ft1/(ft1+ft0)*log(ft1/(ft1+ft0))+
+           ft0/(ft1+ft0)*log(ft0/(ft1+ft0))) +
+                  (ff1+ff0)*(ff1/(ff1+ff0)*log(ff1/(ff1+ff0))+
+           ff0/(ff1+ff0)*log(ff0/(ff1+ff0)))
+ )}
```

Arbre de régression - critère du χ^2

On peut aussi utiliser un test d'indépendance basée sur la distance du χ^2. Posons

$$n_{x,y}^{\perp} = \frac{n_x \cdot n_y}{n}$$

et la distance du χ^2 quand X est partitionné en 2 classes, notées A et B),

$$\chi^2(Y|X) = \sum_{x\in\{A,B\}} \sum_{y\in\{0,1\}} \frac{[n_{x,y} - n_{x,y}^{\perp}]^2}{n_{x,y}^{\perp}}$$

alors que quand X (partitionné en 2 classes, notées A, B et C),

$$\chi^2(Y|X) = \sum_{x\in\{A,B,C\}} \sum_{y\in\{0,1\}} \frac{[n_{x,y} - n_{x,y}^{\perp}]^2}{n_{x,y}^{\perp}}$$

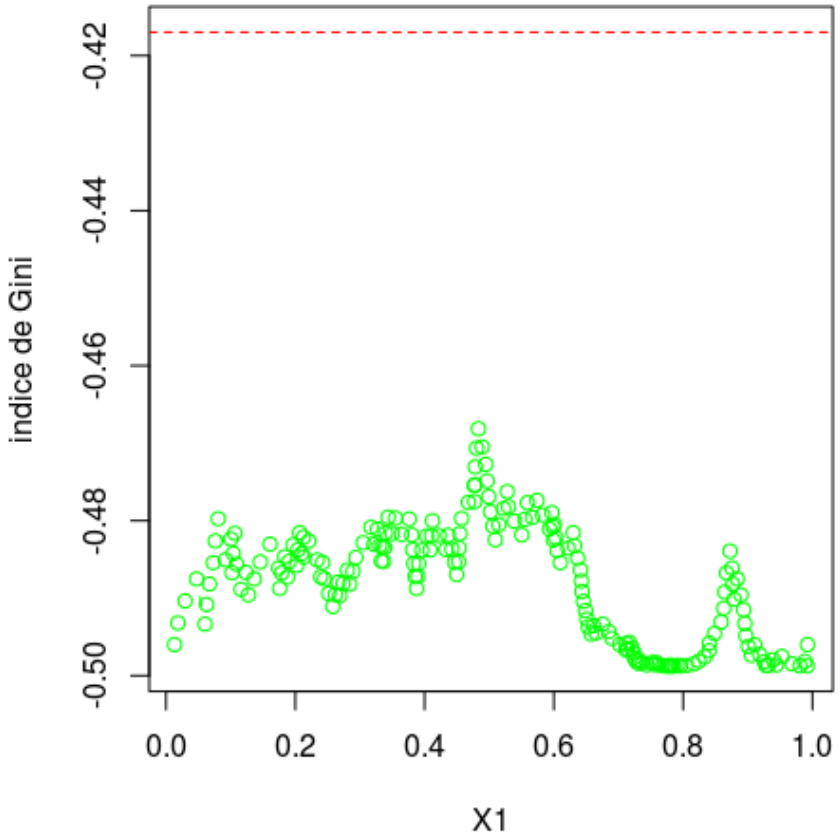
Arbre de régression - critère du χ^2

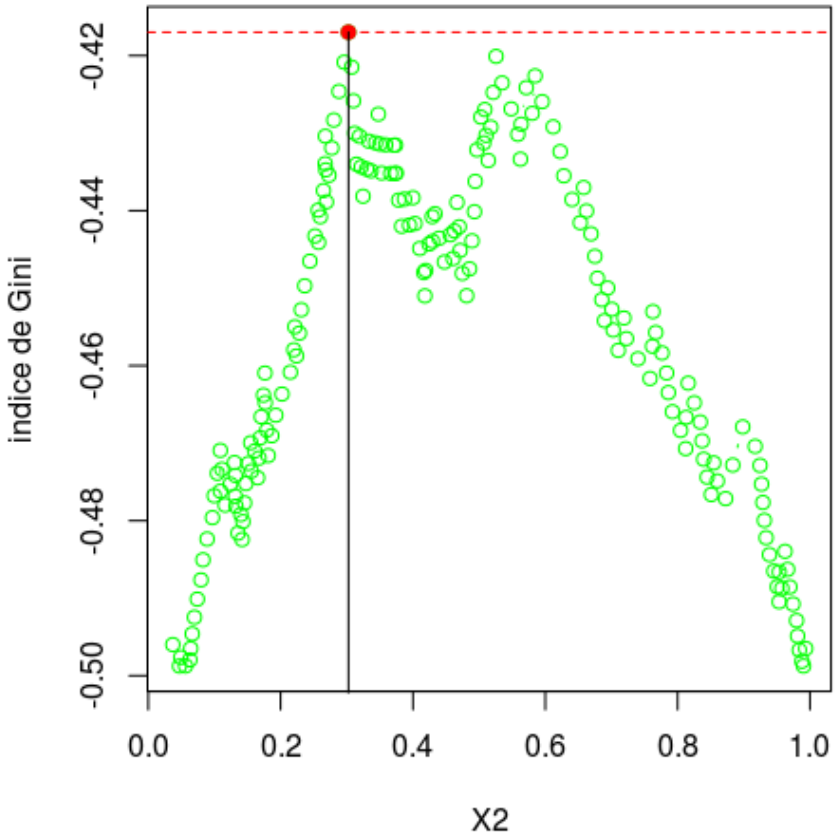
```
> u=sort(unique(B$age))
> gini=entropie=chideux=rep(NA,length(u))
> for(i in 2:(length(u))){
+ I=B$age<(u[i]-.5)
+ ft1=sum(B$Y[I==TRUE])/nrow(B)
+ ff1=sum(B$Y[I==FALSE])/nrow(B)
+ ft0=sum(1-B$Y[I==TRUE])/nrow(B)
+ ff0=sum(1-B$Y[I==FALSE])/nrow(B)
+ M=matrix(c(ft0,ff0,ft1,ff1),2,2)
+ ft=ft0+ft1
+ f0=ft0+ff0
+ Mind=matrix(c(ft*f0,f0*(1-ft),(1-f0)*ft,(1-f0)*(1-ft)),2,2)
+ Q=sum(nrow(B)*(M-Mind)^2/Mind)
+ chideux[i] = Q
+ }
```

Arbre de régression (2 variables)

Considérons deux variables continues $X_{1,i}$ et $X_{2,i}$.

```
> library(tree)
> tree(Y~X1+X2)
```





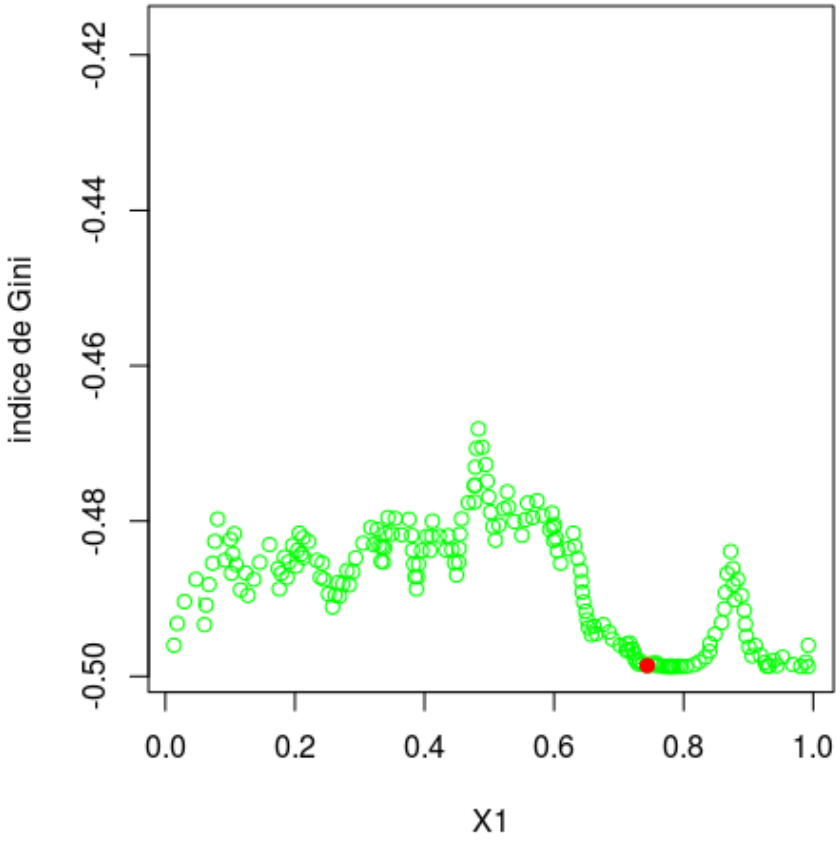
Arbre de régression (2 variables)

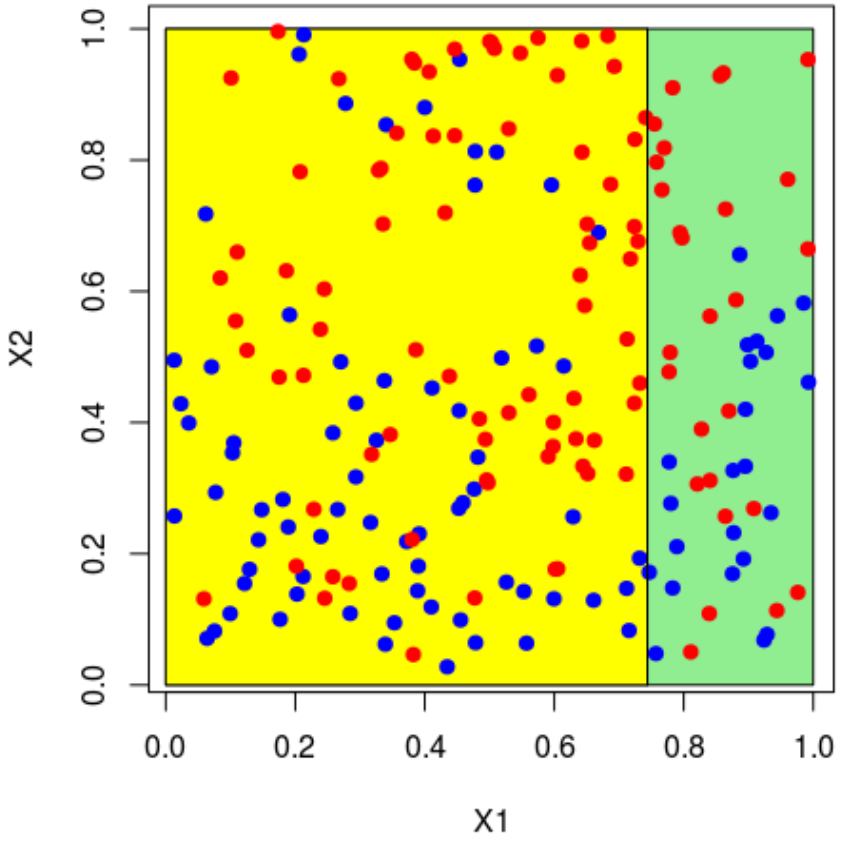
Une animation est présentée sur le blog.

Le principe est le même qu'avec une seule variable, avec un choix de variable (en plus du choix du seuil).

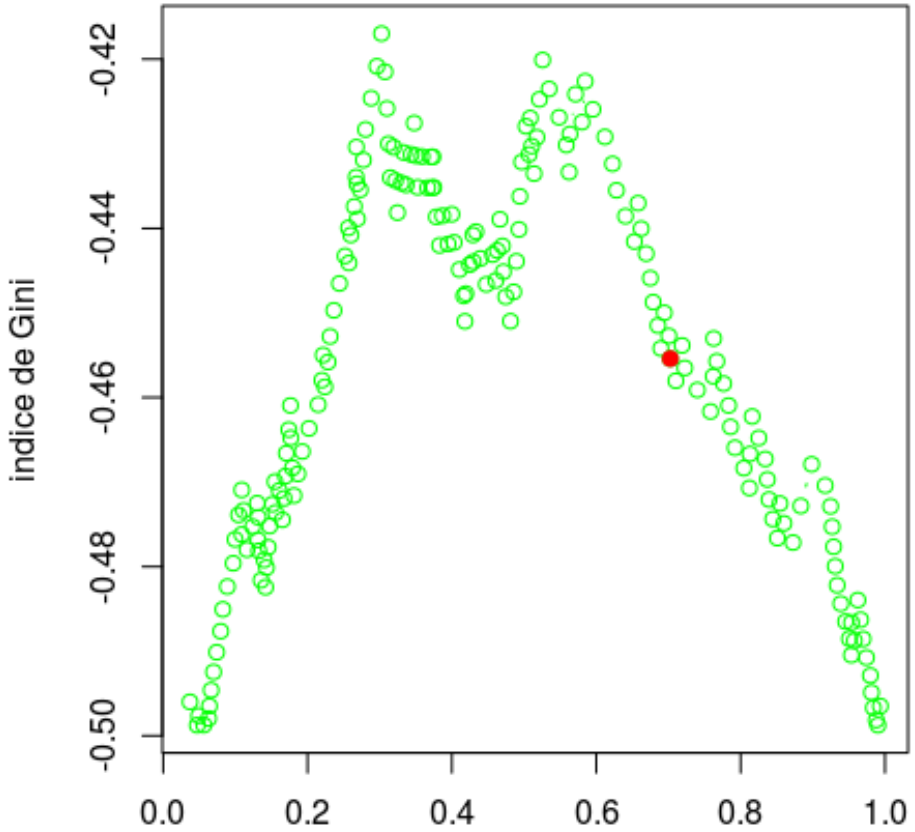
Pour trouver le premier noeud, on calcule l'indice de Gini pour tous les seuils possibles, de X_1 et de X_2.

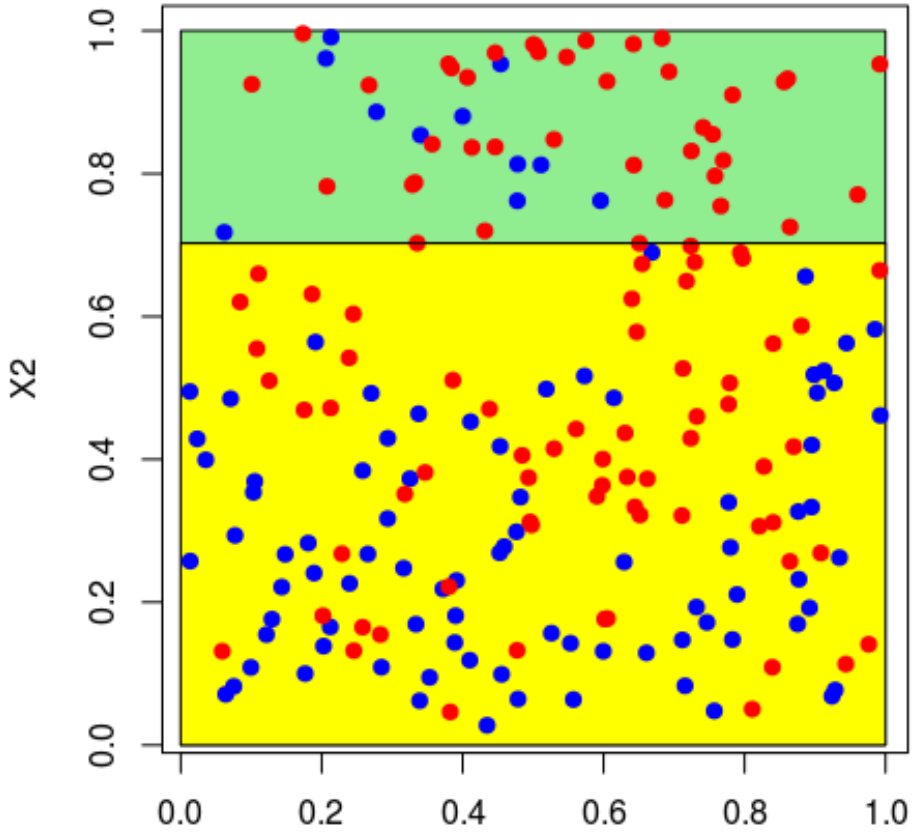
Arbre de régression (2 variables)





Arbre de régression (2 variables)

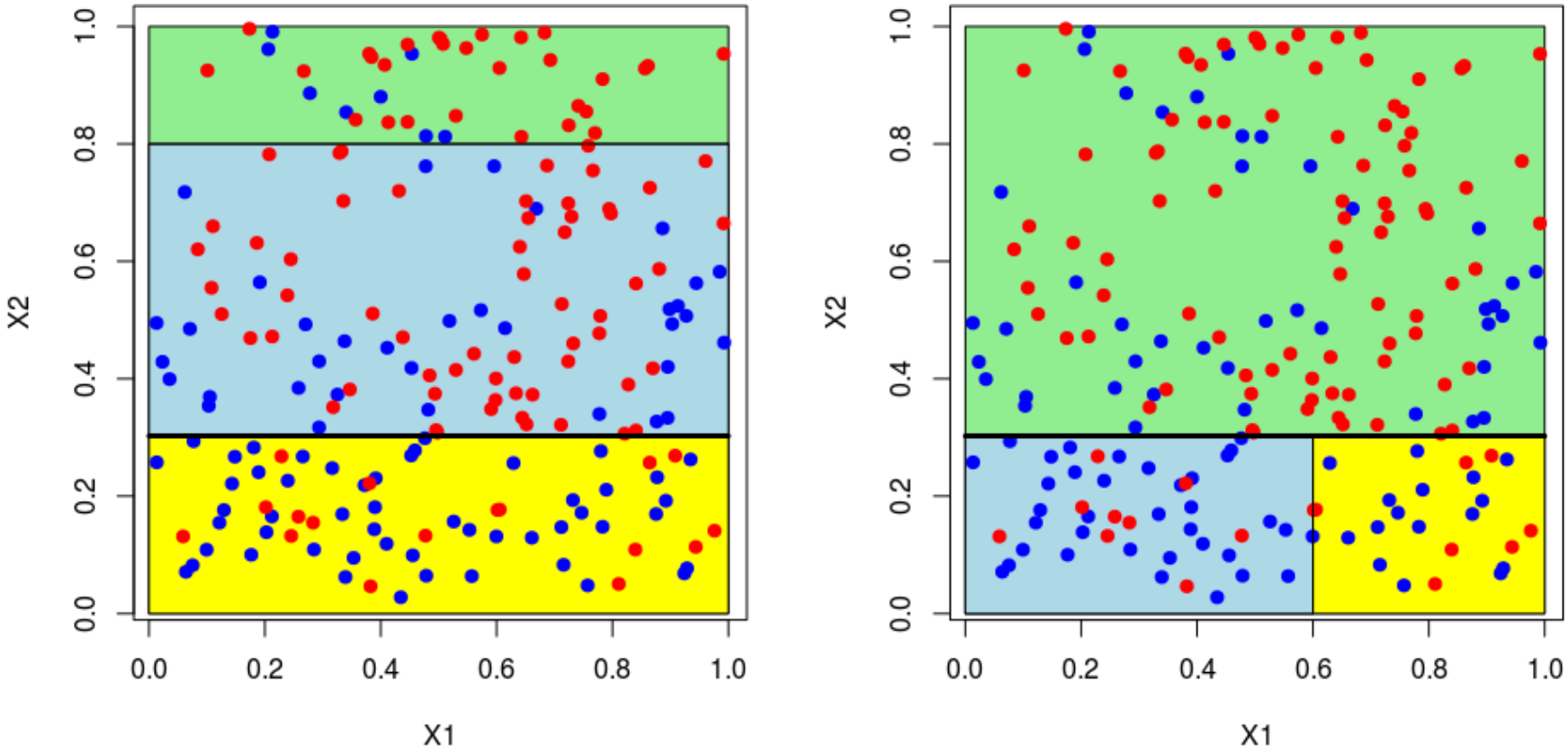




Arbre de régression (2 variables)

Une fois le premier noeud établi (ici sur X_2, en x_2), le second est obtenu

- soit en coupant sur X_1, pour les $X_2 < x_2$
- soit en coupant sur X_1, pour les $X_2 > x_2$
- soit en coupant sur X_2, pour les $X_2 < x_2$
- soit en coupant sur X_2, pour les $X_2 < x_2$



Considérons une variable qualitative $X_{1,i}$.

La variable qualitative prend des modaliés (non ordonnées) $\{x_{1,1}, \cdots, x_{1,I}\}$.

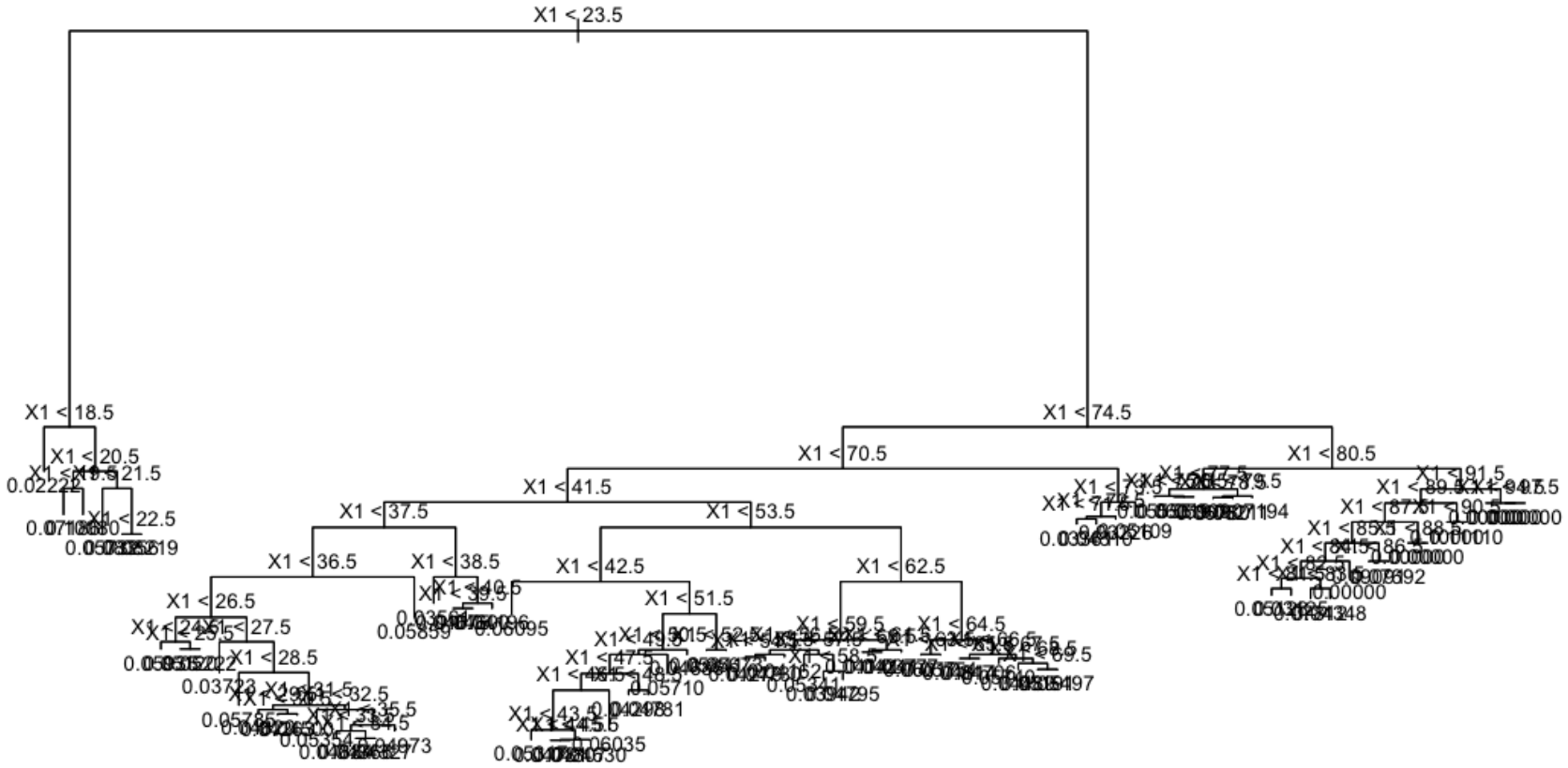
On peut alors calculer des indices de Gini (e.g.) pour toute partition $\{J, J^C\}$ de I.

Couper les arbres

En pratique, sur les grosses bases de données, lire (et interpréter) des arbres peut être difficile,

```
> sinistre=read.table("http://freakonometrics.free.fr/sinistreACT2040.txt",
+ header=TRUE,sep=";")
> sinistres=sinistre[sinistre$garantie=="1RC",]
> contrat=read.table("http://freakonometrics.free.fr/contractACT2040.txt",
+ header=TRUE,sep=";")
> T=table(sinistres$nocontrat)
> T1=as.numeric(names(T))
> T2=as.numeric(T)
> nombre1 = data.frame(nocontrat=T1,nbre=T2)
> I = contrat$nocontrat%in%T1
> T1= contrat$nocontrat[I==FALSE]
> nombre2 = data.frame(nocontrat=T1,nbre=0)
> nombre=rbind(nombre1,nombre2)
> base = merge(contrat,nombre)
> Y = base$nbre>0
> X1 = base$ageconducteur
```

```
> library(tree)
> arbre=tree(Y~X1,split="gini")
> plot(arbre)
> text(arbre,cex=.8)
```



On peut demander à constituer des classes suffisamment larges

```
> arbre=tree(Y~X1,split="gini", mincut = 5000)
> plot(arbre)
> text(arbre,cex=.8)
```