Fiche TD avec le logiciel R : tdr34

Modèles linéaires généralisés

D. Chessel & A.-B. Dufour

Erreur de bernoulli et lien logit, erreur normale et lien identité. Erreur binomiale, erreur poissonienne. Déviances. Modéliser une présence-absence.

Table des matières

1 Introduction : modéliser une probabilité

Supposons qu'on joue à un jeu bizarre dont on fait l'apprentissage au cours d'une série de 20 essais :

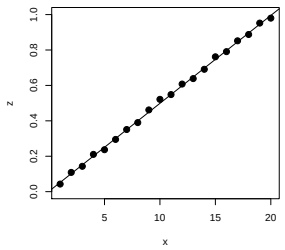
```
x <- 1:20
```

Supposons que la probabilité de gagner croît linéairement de 0.05 le premier coup jusqu'à 1 :

```
y <- x/20
```

Éditer **y**. Mettons une petite erreur sur cette probabilité de gagner :

```
z <- rnorm(rep(1,le=20),y,rep(0.01,le=20))
z[z>0.99] <- 0.98
z[z<0.01] <- 0.01
plot(x,z,ylim=c(0,1),pch=20,cex=2)
abline(lm(z~x))
```



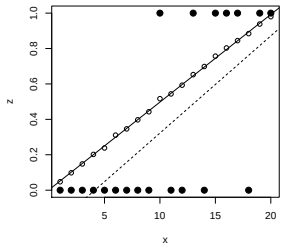
Jusqu'à présent, il n'y a rien de bien extraordinaire. Personne ne connaît la probabilité de gagner. On ne peut qu'observer le résultat (gagné ou perdu).

Fabriquons donc un **résultat observable** de ce modèle :

```
w <- rbinom(rep(1,le=20),rep(1,le=20),z)

plot(x,z,ylim=c(0,1))
abline(lm(z~x))
points(x,w,pch=20,cex=2)
lm1 <- lm(w~x)
abline(lm1,lty=2)
lm1
```

```
Call:
lm(formula = w ~ x)
Coefficients:
(Intercept)            x
   -0.22632      0.05489
```

C'est déjà plus étonnant :

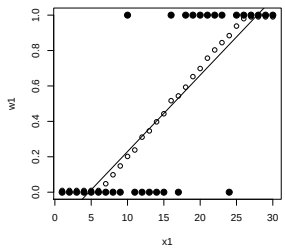


Le modèle $\mathbf{y} = 0.05\ \mathbf{x}$ tient encore si on remplace la probabilité par sa réalisation aléatoire.

Moralité. Les données peuvent être totalement fausses et le modèle accessible. C'est normal, la statistique considère toujours que les données sont " fausses ".

Compliquons un peu. Avant de commencer à apprendre quoi que ce soit, on prend en général quelques baffes. Pendant 6 parties, on ne comprend rien et la probabilité de gagner vaut 0.01. Après l'apprentissage, on a fait le tour de la question et ce n'est plus drôle. Pendant encore 4 parties la probabilité de gagner vaut 0.99 puis on se lasse :

```
x1 <- 1:30
z1 <- c(rep(0.01,le=6),z,rep(0.99,le=4))
z1
 [1] 0.01000000 0.01000000 0.01000000 0.01000000 0.01000000 0.01000000 0.04766567
 [8] 0.09830618 0.14821224 0.20200062 0.23839298 0.31092223 0.34623331 0.39742226
[15] 0.44291965 0.51688291 0.54370808 0.59316436 0.65276363 0.69845222 0.75699590
[22] 0.80330108 0.84466980 0.88428543 0.93820827 0.98000000 0.99000000 0.99000000
[29] 0.99000000 0.99000000
w1 <- rbinom(rep(1,le=30),rep(1,le=30),z1)
w1
 [1] 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 1 1 1 1 1 1 0 1 1 1 1 1 1
plot(x1,w1,pch=20,cex=2,type="p")
abline(lm(w1~x1))
points(x1,z1)
```



Le modèle simple est évidemment invalide sur la réalisation puisque le modèle lui-même n'est pas linéaire. Mais il n'est pas si faux que ça. Il a surtout le défaut de sortir de l'intervalle [0,1]. Une probabilité ne peut être directement une fonction linéaire d'une variable. Pour faire des modèles du type :

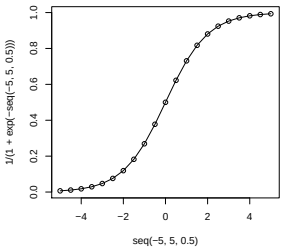
$$\text{probabilité} = \textit{fonction}\ (\text{facteur})$$

un lien s'impose.

2 Erreur de Bernoulli et lien logit

Considérons une variable de milieu x qui varie de 1 à 100 et p la probabilité de rencontrer une espèce donnée qui varie en fonction de x de manière monotone. Il est impossible d'écrire $p = ax + b$ puisque seules les valeurs de l'intervalle $[0, 1]$ ont un sens pour une probabilité. On contourne la difficulté en utilisant la fonction logistique $y = \dfrac{1}{1 + e^{-x}}$:

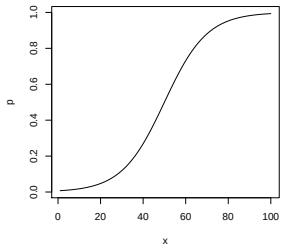
```
plot(seq(-5,5,.5),1/(1+exp(-seq(-5,5,.5))))
lines(seq(-5,5,.5),1/(1+exp(-seq(-5,5,.5))))
```



Considérons donc une variable de milieu x qui varie de 1 à 100 et p la probabilité de rencontrer une espèce donnée qui varie en fonction de x de manière monotone :

$$p = \frac{1}{1 + e^{-0.10x+5}} \Leftrightarrow \log\left(\frac{p}{1-p}\right) = 0.10x - 5$$

```
x <- 1:100
p <- 1/(1+exp(-0.10*x+5))
plot(x,p,type="l")
```

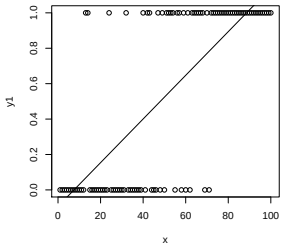


Supposons qu'on échantillonne la présence-absence de l'espèce le long du gradient et que chaque échantillon soit indépendant des autres. Pour une valeur x, la mesure suit une loi de Bernoulli de paramètre $p(x)$. On peut simuler un tel résultat :

```
y1 <- rbinom(100,1,p)
y1
```

```
 [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1
[41] 0 1 1 0 0 0 1 0 1 0 1 1 1 1 0 1 1 0 1 0 1 0 1 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1
[81] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

```
plot(x,y1)
abline(lm(y1~x))
```



L'estimation directe de la probabilité à partir de x n'a pas de sens. Pour estimer les paramètres du modèle avec l'échantillon :

```
glm1 <- glm(y1~x,family=binomial)
glm1
```

```
Call:  glm(formula = y1 ~ x, family = binomial)
Coefficients:
(Intercept)            x
   -4.53920      0.09174

Degrees of Freedom: 99 Total (i.e. Null);  98 Residual
Null Deviance:            138.6
Residual Deviance: 68.8         AIC: 72.8
```

```
pred0.link <- predict(glm1,type="link")
pred0.rep <- predict(glm1,type="response")
```

Le terme **family=binomial** signifie deux choses. La première est que `y1` suit une loi binomiale (pour $n = 1$, donc une loi de Bernoulli), la seconde que ce n'est pas p mais $log(p/(1-p))$ qui est une fonction linéaire de x.

Le " vrai " modèle est défini par $a = 0.10$ et $b = -5$. Le modèle estimé est défini par 0.092 et -4.539.

On peut donc prédire soit le lien, soit la probabilité, l'un dérivant de l'autre. `pred0.link[25]` vaut -2.2457 alors que `pred0.rep[25]` vaut 0.0957.

Exercice. Vérifier à l'aide de la valeur des paramètres la relation qui lie ces deux valeurs.

```
plot(x,y1)
points(x,p,type="l",col="blue",lwd=2)
points(x,predict(glm1,type="response"),pch="+",col="red")
```

Le modèle, les observations et l'estimation sont présents sur la figure 1. On est parti du modèle :

$$Y_i \to B(p_i) \qquad p_i = \frac{1}{1+\exp(-0.10x_i+5)}.$$

On a trouvé l'estimation :

$$p_i = \frac{1}{1+\exp(-0.092x_i+4.539)}$$

.

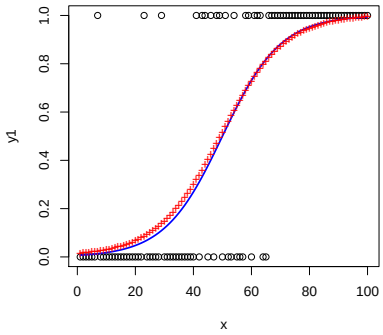


FIGURE 1 : Modèle vrai, données simulées, modèle estimé, erreur binomiale, lien logit.

3 La classe glm

`glm1` est un objet de la classe `glm` :

```
summary(glm1)
Call:
glm(formula = y1 ~ x, family = binomial)
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8109  -0.4737   0.1423   0.4334   2.7990

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.53920    0.90151  -5.035 4.78e-07
x            0.09174    0.01711   5.363 8.18e-08

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 138.589  on 99  degrees of freedom
Residual deviance:  68.802  on 98  degrees of freedom
AIC: 72.802

Number of Fisher Scoring iterations: 5
```

Que signifie la déviance du modèle nul (*Null deviance*) ? Pour comprendre ce terme, on devra impérativement avoir le définition de la vraisemblance. La vraisemblance d'une hypothèse est la probabilité d'observer le résultat obtenu quand cette hypothèse est vraie. On peut réfléchir simplement à ce concept fondamental dans le cas présent car on sait tout sur le modèle (on l'a construit nous-mêmes).

3.1 La vraisemblance des modèles

Nous avons une observation `y1`.

```
y1
  [1] 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
 [41] 1 0 1 1 0 1 0 1 1 0 1 0 0 1 0 0 0 1 1 0 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [81] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

Les observations sont indépendantes. La probabilité d'obtenir ce résultat est le produit des probabilités d'obtenir chaque valeur. $Y_1, Y_2, \cdots, Y_n$ sont des variables aléatoires qui peuvent prendre les valeurs 0 ou 1. $y_1, y_2, \cdots, y_n$ sont les

valeurs observées. Pour chaque modèle possible M, on peut calculer :

$$P_M(Y_1 = y_1, Y_2 = y_2, \cdots, Y_n = y_n) = \prod_{i=1}^{n} P_M(Y_i = y_i)$$

Cette quantité est une probabilité et lorsque M est fixé, c'est une loi de probabilité qui donne la probabilité de tous les résultats possibles. Mais, en statistique, l'observation est donnée définitivement : c'est le résultat de l'observation et bricoler dans les résultats relève de la délinquance scientifique. Devant les résultats fixés, seul le modèle censé les avoir produites peut varier, potentiellement : on parle alors de la fonction de vraisemblance qui devient une fonction du modèle, les résultats étant définitivement fixés :

$$L(M) = \prod_{i=1}^{n} P_M(Y_i = y_i)$$

Un modèle est simplement un ensemble de valeurs qui permet de faire ce calcul. De tels modèles sont nombreux. Le plus cocasse est le modèle *parfait*, qui se dit modèle *complet* (*full model*) ou modèle *saturé* (*satured model*). Un modèle saturé est un modèle pour lequel, la moyenne de la variable observée vaut exactement la valeur observée. Appelons-le S. Y_i suit une loi de Bernoulli de moyenne y_i. Donc quand $y_i = 0$, Y_i prend toujours la valeur 0 et quand $y_i = 1$, Y_i prend toujours la valeur 1. La situation est sans ambiguïté : la probabilité de l'observation est 1. Il n'y a qu'un résultat possible, celui qui est observé.

$$L(S) = \prod_{i=1}^{n} P_M(Y_i = y_i) = 1$$

Le modèle le plus simple est le modèle nul (*null model*). Il est caractérisé par une hypothèse nulle : il n'y a aucun effet du facteur, donc Y_i suit une loi de Bernoulli de moyenne p et p est constante. Il y a donc autant de modèles nuls que de valeurs possibles de p. Les vraisemblances de ces modèles sont donc simples. Si m est le nombre d'observations $y_i = 1$ et, $n - m$ le nombre d'observations $y_i = 0$, alors :

$$L(p) = \prod_{i=1}^{n} P_M(Y_i = y_i) = p^m (1-p)^{(n-m)}$$

Parmi tous les p possibles, on retient celui qui maximise cette fonction. On dit qu'on fait une estimation **MV** (au Maximum de Vraisemblance) ou **MLE** (*Maximal likelihood Estimation*).

Exercice. Dériver cette fonction de p, annuler cette dérivée et observer que l'estimation au maximum de vraisemblance de la probabilité se fait par la fréquence $p_{max} = m/n$, soit 0.51. La probabilité pour que l'espèce soit présente dans un relevé est 0.51. Quelle est la probabilité de l'ensemble des observations dans ce modèle ? On a observé 51 présences de probabilité 0.51 et 49 absences de probabilité 0.49. La vraisemblance est donc :

$$L(p_{max}) = p_{max}^m (1 - p_{max})^{(n-m)}$$

Enfin, une troisième famille de modèles intervient, celle de l'effet x. Elle introduit deux paramètres a et b et la question de la vraisemblance se complique sérieusement. La définition reste :

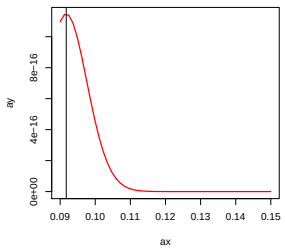
$$L(a,b) = \prod_{y_i=0}\left(1 - \frac{1}{1+\exp(-ax_i-b)}\right)\prod_{y_i=1}\left(\frac{1}{1+\exp(-ax_i-b)}\right)$$

Parmi tous les (a,b) possibles, on retient le couple qui maximise cette fonction. La solution est numérique et produite par la fonction `glm`.

```
a <- glm1$coefficients[2]
b <- glm1$coefficients[1]
p.vec <- 1/(1+exp(-a*x-b))
sum((p.vec-pred0.rep)^2)
```

Toute autre valeur des paramètres donnerait plus (estimation au maximum de vraisemblance) :

```
ax <- seq(0.09,0.15,le=50)
ay <- rep(0,50)
for (i in 1:50) {
    p.vec <- 1/(1+exp(-ax[i]*x-b))
    ay[i] <- prod(p.vec[y1==1])*prod(1-p.vec[y1==0])
}
plot(ax,ay,type="l",lwd=2,col="red")
abline(v=a)
```



3.2 La log-vraisemblance et la déviance résiduelle

Le logarithme de la vraisemblance est donc :

$$\log\left(P\left(observation\right)\right) = LL\left(p\right) = m * log(p) + (n-m)log(1-p)$$

Par définition la log-vraisemblance est -2 fois cette quantité. Le terme déviance désigne une variation de la log-vraisemblance. Le modèle " parfait " est celui où la probabilité de rencontrer l'espèce vaut 1 là où on la rencontre et 0 dans le cas contraire. La vraisemblance de ce modèle est 1 et la log-vraisemblance est nulle. Ceci s'écrit $E\left(Y_i\right) = y_i$. La vraisemblance de l'échantillon est alors définie par $P\left(y_i\right) = 1$ et $2 * LL\left(H\right) = 0$. La variation de vraisemblance entre le modèle parfait et le modèle nul est la déviance résiduelle du modèle nul. Vérifions cette assertion dans le summary.

```
-2*effec*log(proba) -2*(100-effec)*log(1-proba)
```

```
[1] 138.5894
```

Quand on rajoute l'effet du facteur, la probabilité de présence dans la mesure de rang i vaut :

$$P(y_i) = \frac{1}{1 + \exp(-ax_i - b)}$$

La log-vraisemblance de l'échantillon dans le nouveau modèle vaut donc :

$$-2LL(p) = -2 \sum_{y_i=1} \log(P(y_i)) - 2 \sum_{y_i=0} \log(1 - P(y_i))$$

Calculer alors :

```
-2*sum(log(pred0.rep[y1==1]))-2*sum(log(1-pred0.rep[y1==0]))
```

```
[1] 68.80188
```

et retrouver la déviance résiduelle affichée par :

```
glm1
```

```
Call:  glm(formula = y1 ~ x, family = binomial)
Coefficients:
(Intercept)            x
   -4.53920      0.09174

Degrees of Freedom: 99 Total (i.e. Null);  98 Residual
Null Deviance:            138.6
Residual Deviance: 68.8         AIC: 72.8
```

La vraisemblance du modèle saturé est 1 donc sa log-vraisemblance est 0. Il y a dans ce cas (et dans ce cas seulement) confusion entre variation de vraisemblance par rapport au modèle saturé et -2*log-vraisemblance puisqu'on part de 0. La déviance de chaque modèle se réfère à la même valeur, ici 0 (c'est un avantage pédagogique mais une complication statistique). Le modèle nul donne une déviance de 138.589 avec estimation d'un seul paramètre p, le modèle en **x** donne une déviance de 68.802 avec une estimation de deux paramètres a et b. Les estimations sont au maximum de vraisemblance et l'ensemble des valeurs du premier (une constante) est contenu dans l'ensemble des valeurs du second. La différence des déviances est donc un rapport de vraisemblance et, si le premier modèle est vrai, cette quantité suit un χ^2 à un degré de liberté :

```
anova(glm1,test="Chisq")
```

```
Analysis of Deviance Table
Model: binomial, link: logit

Response: y1

Terms added sequentially (first to last)


     Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                    99    138.589
x     1   69.788        98     68.802 < 2.2e-16
```

3.3 Un autre exemple

C'est un avantage décisif de R. Pour comprendre un modèle, l'expérimentateur peut en faire une réalisation. La seconde ne peut remplacer le premier mais donne des idées sur la façon dont les données sont censées être obtenues. L'exemple est de Ter Braak et Looman [3]. x est une variable de milieu, **y** l'observation de la présence (1) ou l'absence (0) d'une espèce, le couple des deux est un archétype de *courbe de réponse* de l'espèce à la variable.

```
x <- c(20,23,26,30,33,36,40,43,46,50,53,56,60,70,80,90)
x2 <- x*x
y<- c(0,0,0,0,0,0,0,0,0,1,1,0,1,1,0,0)
```

- ⋆ Le modèle 'saturé' est parfait, sa log-vraisemblance est nulle.
- ⋆ Le modèle 'nul' considère qu'il n'y a pas d'effet, la fréquence observée est 1/4, la log-vraisemblance est $-2 * log(0.75^{12} * 0.25^{4})$ soit 17.9947.
- ⋆ Le modèle 'effet linéaire' conduit à une estimation au maximum de vraisemblance de a=0.03783 et b=-3.01026.

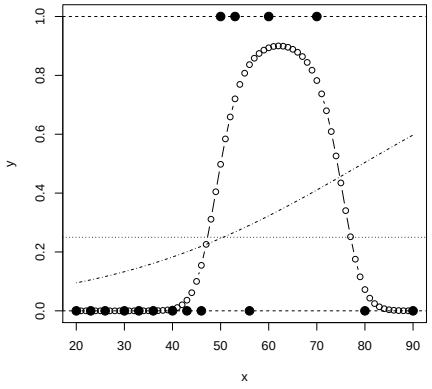
Supposons que ce dernier modèle soit vrai. La probabilité de rencontrer l'espèce est donc :

```
probabilite
    1     2     3     4     5     6     7     8     9    10    11    12    13    14
0.095 0.105 0.116 0.133 0.147 0.161 0.183 0.200 0.219 0.246 0.268 0.291 0.323 0.410
   15    16
0.504 0.597
```

Exercice. Retrouvez ces valeurs. Aidez-vous de la fonction ci-dessous pour faire l'expérience d'un échantillon de la réponse conformément à ce modèle pour lequel on estime la déviance associée à la variable **x2** dont on sait (par construction) qu'elle n'a aucun effet.

```
f1 <- function (k=1) {
    sim <-  rbinom(proba,1,prob=proba)
    glmsim<-glm(sim~x+x2)
    w <- anova(glmsim,test="Chisq")[3,2]
    w
}
```

Faites alors 500 fois cette expérience et confrontez la distribution observée à une loi χ^2_1. Faites le test sur les observations initiales. S'agissant d'application de théorèmes d'approximation, l'approximation n'est pas toujours très bonne, en particulier pour des effectifs petits. Les données en présence-absence sont de maniement délicat. Achevez en retrouvant la figure :



4 Erreur normale et lien identité

Le modèle linéaire généralisé contient deux éléments fondamentaux.

4.1 Le premier est le type d'erreur.

Dans un modèle linéaire (figure 2), elle est normale de variance constante, par exemple :

```
x <- 1:100
yvrai <- 0.025*x+0.075
ysim <- rnorm(100,sd=1) + yvrai
lmnorm <- lm(ysim~x)
plot(x,ysim,type = "p")
abline(c(0.075,0.025), col="blue", lwd=2)
points(x,lmnorm$fitted.values, pch="+", col="red")
legend(0,max(ysim),c("modèle vrai","modèle estimé"),col=c("blue","red"),pch=c("-","+"))
```

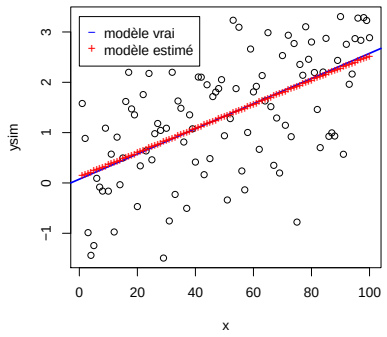


FIGURE 2 : Modèles et données simulées, erreur normale et lien identité

Le modèle linéaire est aussi un modèle linéaire généralisé :

```
glmnorm <- glm(ysim~x,family=gaussian)
glmnorm
```

```
Call:  glm(formula = ysim ~ x, family = gaussian)
Coefficients:
(Intercept)            x
    0.02997      0.02559

Degrees of Freedom: 99 Total (i.e. Null);  98 Residual
Null Deviance:            163.2
Residual Deviance: 108.6         AIC: 298.1
```

```
lmnorm
```

```
Call:
lm(formula = ysim ~ x)
Coefficients:
(Intercept)            x
    0.02997      0.02559
```

```
anova(lmnorm)
```

```
Analysis of Variance Table
Response: ysim
          Df  Sum Sq Mean Sq F value    Pr(>F)
x          1  54.583  54.583  49.236 2.966e-10
Residuals 98 108.641   1.109
```

```
anova(glmnorm,test="F")
```

```
Analysis of Deviance Table
Model: gaussian, link: identity

Response: ysim

Terms added sequentially (first to last)
```

```
     Df Deviance Resid. Df Resid. Dev      F    Pr(>F)
NULL                    99     163.22
x     1   54.583        98     108.64 49.236 2.966e-10
```

4.2 Le second est la fonction de lien.

Dans le modèle linéaire, on cherche directement la liaison sous la forme $y = ax + b$. Dans le modèle linéaire généralisé, on cherche à prédire la fonction de lien sous la forme :

$$\log\left(\frac{p}{1-p}\right) = ax + b \Leftrightarrow p = \frac{1}{1+e^{-(ax+b)}}$$

Un modèle linéaire est un modèle linéaire généralisé d'erreur normale et de lien identité.

5 Erreur binomiale

Si pour chaque valeur de **x**, on pouvait faire 5 mesures indépendantes :

```
x <- 1:100
p <- 1/(1+exp(-0.10*x+5))
y3 <- rbinom(100,5,p)
plot(x,y3/5)
lines(x,p,col="blue",lwd=2)
glm3 <- glm(cbind(y3,5-y3)~x,family=binomial)
glm3
```

```
Call:  glm(formula = cbind(y3, 5 - y3) ~ x, family = binomial)
Coefficients:
(Intercept)            x
    -5.1727       0.1033

Degrees of Freedom: 99 Total (i.e. Null);  98 Residual
Null Deviance:            450.2
Residual Deviance: 67.72        AIC: 153.2
```

```
points(x,predict(glm3,type="response"),pch="+",col="red")
legend(0,max(y3/5),c("modèle vrai","modèle estimé"),col=c("blue","red"),pch=c("-","+"))
```

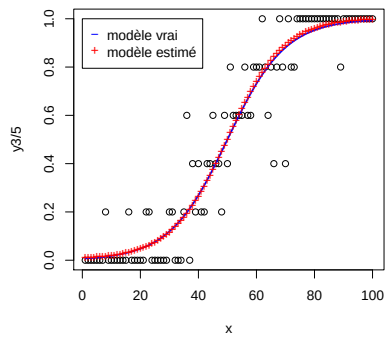


FIGURE 3 : Modèles et données simulées ; erreur binomiale et lien logit

Bien noter la syntaxe :

`glm(cbind(nsucces,nechec)~x,family=binomial)`

En chaque point de **x**, on fait 5 essais (figure 3). La probabilité de succès est $p(x)$. On obtient un résultat entre 0 à 5, tirage d'une loi binomiale de paramètre 5 et $p(x)$, qui donne des fréquences observées possibles 0, 0.2, 0.4, 0.6, 0.8 et 1. On utilise le même lien mais l'erreur est binomiale.

6 Erreur de Poisson

Supposons enfin que pour chaque valeur de **x**, on compte un nombre d'individus (figure 4). Ce nombre suit une loi de Poisson de paramètre m vérifiant $\log(m) = ax + b$. L'erreur a changé, elle est poissonienne. On change aussi le lien :

```
x <- 1:100
m <- exp(0.025*x+0.075)
y4 <- rpois(100,m)
plot(x,y4)
lines(x,m,col="blue",lwd=2)
glm4 <- glm(y4~x,family=poisson)
glm4
```

```
Call:  glm(formula = y4 ~ x, family = poisson)
Coefficients:
(Intercept)            x
    0.18835      0.02323

Degrees of Freedom: 99 Total (i.e. Null);  98 Residual
Null Deviance:              307.5
Residual Deviance: 115.6        AIC: 423.9
```

```
points(x,predict(glm4,type="response"),pch="+",col="red")
legend(0,max(y4),c("modèle vrai","modèle estimé"),col=c("blue","red"),pch=c("-","+"))
```

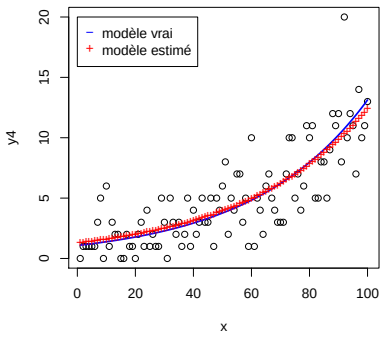


FIGURE 4 : Modèles et données simulées ; erreur poissonienne et lien exponentiel.

En chaque point de x, le résultat est un entier réalisation d'une loi de Poisson de paramètre $m(x)$. On a obtenu 0.024 et -0.013 pour 0.025 et 0.075. Les résultats sont toujours bons parce que les hypothèses du modèle sont satisfaites et que l'échantillonnage est régulier sur le gradient. La fonction `glm` permettra de passer de la régression simple à la régression logistique ou poissonienne. Il y a d'autres liens possibles et d'autres erreurs possibles. Voir `help(family)` pour en avoir une idée.

7 Le fonctionnement de la régression logistique

On emprunte au cours de Jacques Estève sur le modèle linéaire généralisé (4 octobre 2006) cette illustration du fonctionnement du calcul dans une régression logistique.

Âge, Tabac et probabilité de 'succès' !

```
age <- c(25.0,32.5,37.5,42.5,47.5,52.5,57.5,65.0)
n <- c(100,150,120,150,130,80,170,100)
Y <- c(10,20,30,50,60,50,130,80)
```

`Y` est le nombre d'individus présentant des symptômes cardio-vasculaires, `n` est le nombre d'essais et `age` une variable explicative quantitative. L'approximation numérique de la régression logistique est :

```
f <- Y/n
f
```
```
[1] 0.1000000 0.1333333 0.2500000 0.3333333 0.4615385 0.6250000 0.7647059 0.8000000
```

Transformons les données par le lien :

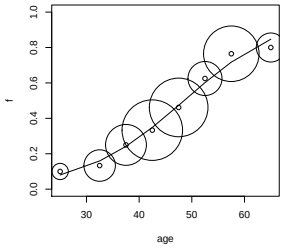
```
g <- log(f/(1-f))
```

Pondérons les données :

```
w <- n*f*(1-f)
```

Faisons la régression pondérée et la transformation inverse :

```
r <- predict(lm(g~age,weights=w))
p <- exp(r)/(1+exp(r))
p
```
```
        1         2         3         4         5         6         7
0.08006599 0.15935489 0.24156976 0.34861123 0.47347343 0.60174404 0.71741610
        8
0.84684866
```
```
plot(age,f,ylim=c(0,1))
lines(age,p)
symbols(age,f,circles=w,add=T,inc=0.5)
```



On peut itérer cette régression pondérée :

```
p
```
```
        1         2         3         4         5         6         7
0.08006599 0.15935489 0.24156976 0.34861123 0.47347343 0.60174404 0.71741610
        8
0.84684866
```

```
w <- n*p*(1-p)
gu <- r+(f-p)/p/(1-p)
r <- predict(lm(gu~age,weights=w))
r
         1          2          3          4          5          6          7
-2.4651069 -1.6775179 -1.1524586 -0.6273992 -0.1023399  0.4227194  0.9477787
         8
 1.7353677
p <- exp(r)/(1+exp(r))
p
         1          2          3          4          5          6          7
0.07834081 0.15742442 0.24004030 0.34810049 0.47443733 0.60413380 0.72066824
         8
0.85009772
```

Encore une fois :

```
w <- n*p*(1-p)
gu <- r+(f-p)/p/(1-p)
r <- predict(lm(gu~age,weights=w))
p <- exp(r)/(1+exp(r))
p
         1          2          3          4          5          6          7
0.07833012 0.15741201 0.24002985 0.34809573 0.47444116 0.60414616 0.72068596
         8
0.85011588
```

On peut montrer qu'on obtient ainsi les estimations au maximum de vraisemblance. :

```
coefficients(lm(gu~age,weights=w))
(Intercept)         age
 -5.0907332   0.1050191
coefficients(glm(cbind(Y,n-Y)~age,family=binomial))
(Intercept)         age
 -5.0907332   0.1050191
predict(glm(cbind(Y,n-Y)~age,family=binomial),type="response")
         1          2          3          4          5          6          7
0.07833012 0.15741201 0.24002985 0.34809573 0.47444116 0.60414616 0.72068596
         8
0.85011588
```

Exercice. Vérifier l'assertion dans ce cas numérique. La convergence est-elle rapide ?

8 Modéliser une présence-absence

8.1 Le problème

L'exercice est basé sur un extrait des bases de données du Programme National " Indice Poisson ". GIP Hydrosystèmes, CSP, Agences de Bassin. Mise au point d'un indice Poisson applicable sur le territoire national : Convention n° 1302 Conseil Supérieur de la pêche / Agence de l'eau Adour-Garonne. Août 1996 - Décembre 2000. Responsable scientifique : T. Oberdorff [2]. Installer le data.frame `gardonmi` en utilisant le fichier :

`http://pbil.univ-lyon1.fr/R/donnees/gardonmi.txt`

Lister les 10 premières lignes :

```
gardonmi[1:10,]
```

```
      S     V     G     A gardon
1  Nord -0.49  0.31 -1.26      0
2  Nord -0.56  0.79 -1.65      0
3  Nord -0.59 -0.17 -0.22      1
4  Nord -1.30 -1.26 -1.17      0
5  Nord  0.30  0.36  0.15      1
6  Nord -0.65  0.77  0.06      1
7  Nord -0.03 -1.38  0.18      0
8  Nord -1.12  1.76  0.17      1
9  Nord -0.22 -1.21  0.40      0
10 Nord -0.52  0.94  0.17      1
```

```
dim(gardonmi)
```

```
[1] 645   5
```

Dans 645 stations de référence, le Conseil Supérieur de la Pêche (CSP) a enregistré la présence ou l'absence du Gardon , *Rutilus rutilus* L., ce qui donne la variable `gardon` en 0-1. Chaque station appartient à un bassin (variable S) :

```
levels(gardonmi$S)
```

```
[1] "Atla" "Garo" "Loir" "Manc" "Medi" "Nord" "Rhon" "Sein"
```

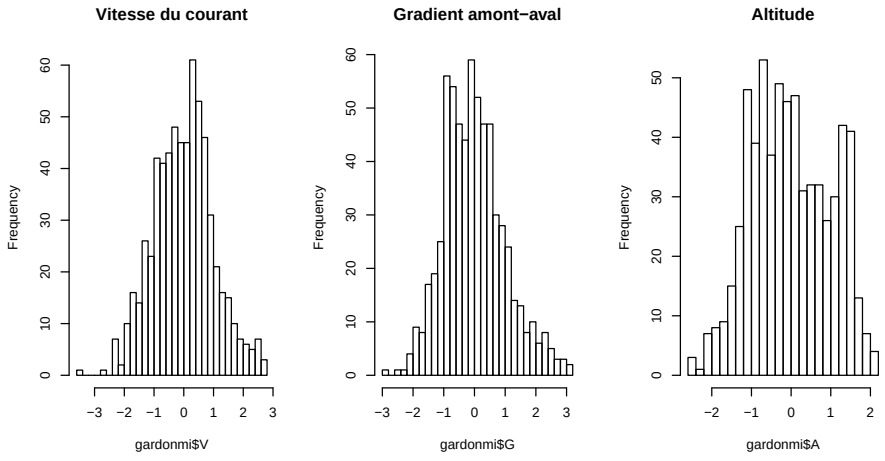
```
summary(gardonmi$S)
```

```
Atla Garo Loir Manc Medi Nord Rhon Sein
  56  103   84   74   69   42  102  115
```

On connaît pour chaque station un indice caractérisant les conditions hydrauliques locales basé sur la vitesse du courant, la pente et la largeur (`V`, variable normalisée), la position de la station dans le gradient Amont-Aval basée sur la distance à la source et la surface du bassin draîné (`G`, variable normalisée) et l'altitude (`A`, variable transformée et normalisée).

Les distributions des variables explicatives sont convenables :

```
par(mfrow=c(1,3))
hist(gardonmi$V,nclass=25, main="Vitesse du courant")
hist(gardonmi$G,nclass=25, main="Gradient amont-aval")
hist(gardonmi$A,nclass=25, main="Altitude")
```

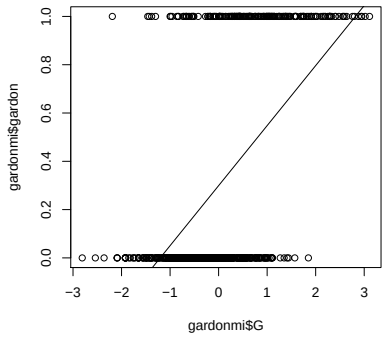


L'objectif est de modéliser la présence du Gardon par les variables environnementales. La contrainte provient de la variable à expliquer (0-1) dont les valeurs sont des réalisations d'un événement qui avait une certaine probabilité de survenir. On ne modélise donc pas le résultat mais la probabilité de ce résultat, de

même qu'on ne modélise pas une valeur observée mais la moyenne des valeurs observées dans les mêmes conditions :

$$X = E(X) + erreur \quad et \quad E(X) = f(\vartheta)$$

```
plot(gardonmi$G,gardonmi$gardon)
abline(lm(gardonmi$gardon~gardonmi$G))
```



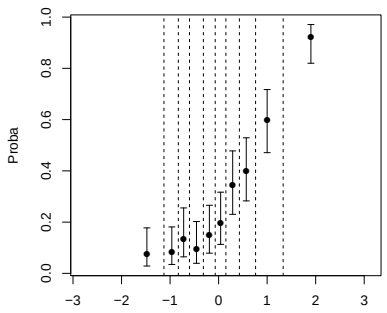
Ce dont on a besoin s'exprime clairement par la fonction suivante dont on détaillera les constituants. Récupérer le fichier :

`http://pbil.univ-lyon1.fr/R/donnees/plot.freq.grad.R`

Lister la fonction après intervention de la fonction `source`, identifier le contenu (il n'y a que des opérations simples) et tracer le graphe :

```
source("http://pbil.univ-lyon1.fr/R/donnees/plot.freq.grad.R")
plot.freq.grad
function (xobs , yobs , ncla = 10)
{
if (is.numeric(xobs) == F) return("numeric expected")
q0 <- quantile(xobs, probs = seq(0, 1, 1/ncla), na.rm = F)
c.cla <- quantile(xobs, probs = seq(0 + 1/2/ncla, 1 - 1/2/ncla, 1/ncla), na.rm = F)
xmin <- min(xobs) ; xmax <- max(xobs)
q1 <- cut(xobs, q0, include.lowest = T)
t0 <- table(q1, yobs)
t0[, 1] <- (t0[, 1] + t0[, 2])
freq <- t0[, 2]/t0[, 1]
basbar <- rep(0, ncla) ; haubar <- rep(0, ncla)
for (i in 1:ncla) {
succes <- t0[i, 2] ; essai <- t0[i, 1]
if (essai > 10) {
a0 <- prop.test(succes, essai)$conf.int
basbar[i] <- a0[1] ; haubar[i] <- a0[2]
if (a0[1] > freq[i]) basbar[i] <- NA
if (a0[2] < freq[i]) haubar[i] <- NA
} else {
basbar[i] <- NA ; haubar[i] <- NA
}
}
ymin <- min(basbar, na.rm = T)
ymin <- min(ymin,min(freq))
ymax <- max(haubar, na.rm = T)
ymax <- max(ymax,max(freq))
plot(0, 0, ylim = c(ymin, ymax), xlim = c(xmin, xmax), xlab="", ylab = "Proba", type = "n")
points(c.cla, freq, pch = 16)
```

```
for (i in 1:ncla) {
if (!is.na(basbar[i])) {
size.bar <- par("cxy")[1]/4
segments(c.cla[i], basbar[i] , c.cla[i], haubar[i] )
segments(c.cla[i] - size.bar, haubar[i] , c.cla[i] + size.bar, haubar[i] )
segments(c.cla[i] - size.bar, basbar[i] , c.cla[i] + size.bar, basbar[i] )
}
if (i > 1) abline(v = q0[i], lty = 2)
}
}
plot.freq.grad(gardonmi$G,gardonmi$gardon)
```



Si la probabilité de rencontrer du Gardon dans une station de milieu ϑ s'écrit :

$$P(X=1) = E(X) = f(\vartheta)$$

on se trompe toujours fortement dans l'observation ! On demande à estimer cette probabilité de manière qu'en moyenne pour les stations de probabilité 0.15 on ait du Gardon dans 15% des cas. Autour de la prévision (probabilité), l'observation (oui/non) réalise une erreur très particulière dite de type binomiale (cas particulier de Bernoulli, $n = 1$). La fonction d'erreur est la première généralisation du modèle linéaire (erreur non gaussienne). La seconde fait que cette probabilité (comprise entre 0 et 1) ne peut être une fonction linéaire des paramètres. On ne prédit pas linéairement p mais une fonction inversible de p. Le modèle s'écrit :

$$\begin{aligned} g(P(X=1)) &= f_{lin}(\vartheta) \\ P(X=1) &= g^{-1}(f_{lin}(\vartheta)) \end{aligned}$$

g, dite fonction de lien est la seconde généralisation du modèle linéaire. On parle alors de modèles linéaires généralisés (**glm**). On peut utiliser le lien logit qui par inversion renvoie à la fonction logistique :

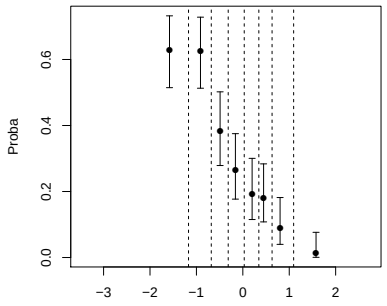
$$g(p) = \log\left(\frac{p}{1-p}\right) = f(\vartheta) \Leftrightarrow p = \frac{1}{1+e^{-f(\vartheta)}}$$

Dans R, un seul paramètre dans `glm` suffit à se mettre dans cette situation. Lire la documentation de la fonction `glm`.

8.2 Courbes de réponse

La probabilité de présence du Gardon dépend-elle de la vitesse ?

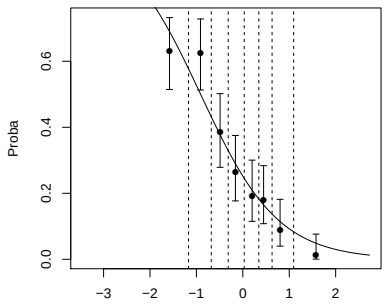
```
plot.freq.grad(gardonmi$V,gardonmi$gardon,8)
```




```
gardon <- gardonmi$gardon
V <- gardonmi$V
glm1 <- glm(gardon~V, family=binomial)
glm1
```

```
Call:  glm(formula = gardon ~ V, family = binomial)
Coefficients:
(Intercept)            V
     -1.088       -1.196

Degrees of Freedom: 644 Total (i.e. Null);  643 Residual
Null Deviance:            787.2
Residual Deviance: 652          AIC: 656
```

```
xnou <- seq(min(V),max(V),len=100)
ynou <- predict(glm1,data.frame(V=xnou),type="response")
plot.freq.grad(V,gardon,8)
lines(xnou,ynou)
```

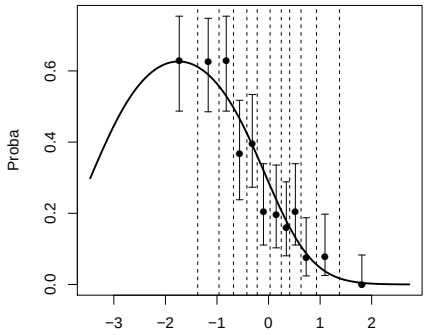


Doit-on ajouter un terme carré ?

```
gardon <- gardonmi$gardon
V <- gardonmi$V
glm2 <- glm(gardon~V+I(V^2), family=binomial)
anova(glm2,test="Chisq")
```

```
Analysis of Deviance Table
Model: binomial, link: logit

Response: gardon

Terms added sequentially (first to last)


       Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                     644     787.17
V       1  135.122       643     652.04 < 2.2e-16
I(V^2)  1   16.756       642     635.29 4.251e-05
```

```
plot.freq.grad(V,gardon,12)
lines(xnou,predict(glm2,data.frame(V=xnou),type="response"),lwd=2)
```

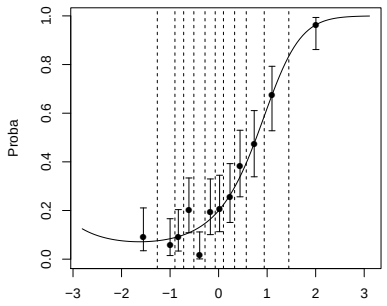


Noter l'usage de I() :

While formulae usually involve just variable and factor names, they can also involve arithmetic expressions. The formula 'log(y) a + log(x)' is quite legal. When such arithmetic expressions involve operators which are also used symbolically in model formulae, there can be confusion between arithmetic and symbolic operator use. To avoid this confusion, the function 'I()' can be used to bracket those portions of a model formula where the operators are used in their arithmetic sense. For example, in the formula 'y a + I(b+c)', the term 'b+c' is to be interpreted as the sum of 'b' and 'c'.

La probabilité de présence du Gardon dépend-elle aussi du gradient Amont-Aval ?

```
G <- gardonmi$G
glm3 <- glm(gardon~G+I(G^2),family=binomial)
plot.freq.grad(G,gardon,12)
xnou <- seq(min(G),max(G),len=100)
lines(xnou,predict(glm3,data.frame(G=xnou),type="response"))
```



8.3 Surfaces de réponse

Écrire le modèle :

```
glm4 <- glm(gardon~V+G+I(V^2)+I(G^2)+I(V*G), family=binomial)
anova(glm4,test="Chisq")
```

```
Analysis of Deviance Table
Model: binomial, link: logit

Response: gardon

Terms added sequentially (first to last)


         Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                       644     787.17
V         1  135.122       643     652.04 < 2.2e-16
G         1  135.313       642     516.73 < 2.2e-16
I(V^2)    1    9.071       641     507.66 0.0025968
I(G^2)    1   11.213       640     496.45 0.0008122
I(V * G)  1    2.116       639     494.33 0.1457687
```

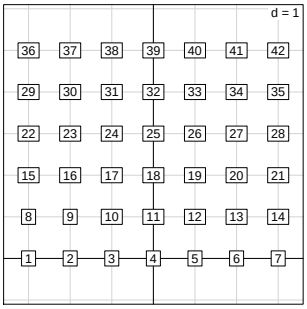
```
glm5 <- glm(gardon~V+G+I(V^2)+I(G^2), family=binomial)
summary(glm5)
```

```
Call:
glm(formula = gardon ~ V + G + I(V^2) + I(G^2), family = binomial)
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9889  -0.6443  -0.3281   0.3362   2.9809

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -1.3705     0.1549  -8.849  < 2e-16
V            -1.3262     0.2124  -6.245 4.23e-10
G             1.1515     0.1401   8.216  < 2e-16
I(V^2)       -0.4079     0.1412  -2.889 0.003862
I(G^2)        0.3840     0.1148   3.346 0.000821

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 787.17  on 644  degrees of freedom
Residual deviance: 496.45  on 640  degrees of freedom
AIC: 506.45

Number of Fisher Scoring iterations: 6
```

Pour voir ce qu'il se passe, de nombreuses techniques sont disponibles.

Mettre en place une grille de valeurs des prédicteurs. Identifier d'abord ce que fait la fonction `expand.grid` :

```
s.label(expand.grid(seq(-3,3,by=1),0:5))
```

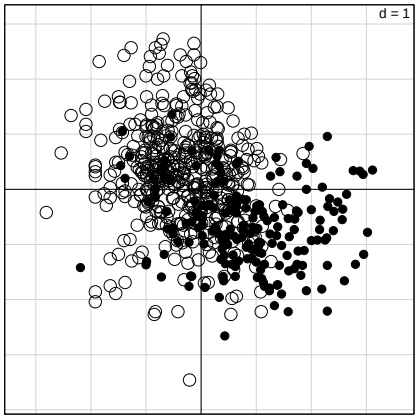


Estimer le modèle pour les valeurs de la grille :

```
newG <- seq(min(G),max(G),le=20)
newV <- seq(min(V),max(V),le=20)
newdata <- expand.grid(G=newG, V=newV)
newresult <- predict.glm(glm4,newdata,type="response")
```

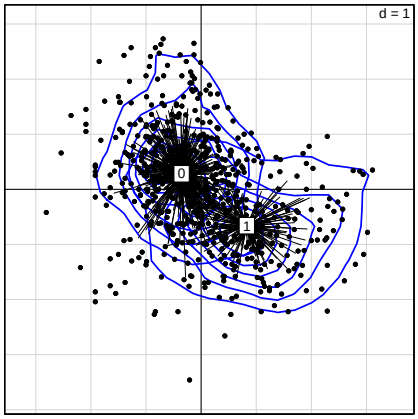
Confronter les données et le modèle :

```
s.label(newdata,clab=0,cpoi=0)
s.label(data.frame(G,V),pch=c(1,20)[gardon+1],cpoi=2,clab=0,add.p=T)
```



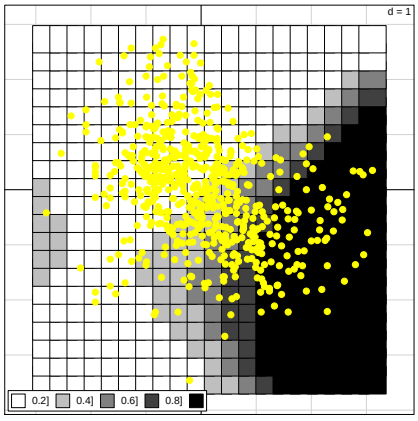
Ici, on a les observations brutes avec en abscisse, le gradient G (à gauche l'amont des rivières, à droite l'aval) et en ordonnée la vitesse de l'eau. Les points blancs donnent les absences, les points noirs donnent les présences. Deux observations valident la représentation. Les eaux calmes se rencontrent dans les parties avales et les deux prédicteurs ne sont pas indépendants (calculer leur corrélation). Le Gardon est une espèce limnophile et préfère les eaux lentes. Enrichir la lecture des données :

```
if (require(MASS)) {
    s.label(newdata,clab=0,cpoi=0)
    s.kde2d(data.frame(G,V)[gardon==0,],add.p=T)
    s.kde2d(data.frame(G,V)[gardon==1,],add.p=T)
    s.class(data.frame(G,V),factor(gardon),csta=0.5,cell=0,add.p=T)
}
```

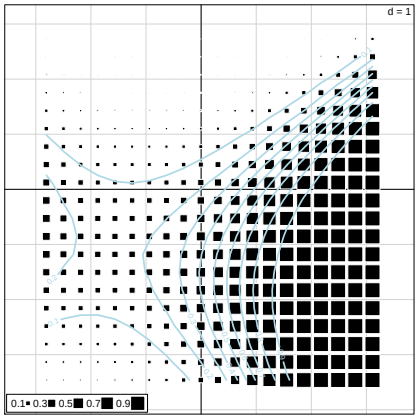


La représentation montre l'ensemble des points où on trouve du Gardon et l'ensemble des points où on n'en trouve pas. Il serait ridicule de tester l'existence d'une niche écologique mais la discussion sur le rôle de chacune des variables n'est pas sans intérêt.

```
old=par("mar")
par(mar=c(0.1,0.1,0.1,0.1))
s.value (newdata,newresult,meth="greylevel")
points(G,V,pch=19,col="yellow")
par(mar=old)
```

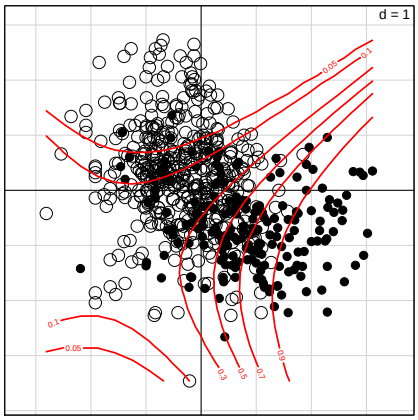



```
old=par("mar")
par(mar=c(0.1,0.1,0.1,0.1))
s.value (newdata,newresult,csi=0.55)
contour(newG,newV,matrix(newresult,20),xlab="G",ylab="V",add=T,lwd=2,col="lightblue")
par(mar=old)
```



Ici on ne voit que le modèle, les valeurs estimées sur la grille et les courbes de niveaux de la surface qui exprime le modèle. On peut enfin croiser les données et le modèle :

```
old=par("mar")
par(mar=c(0.1,0.1,0.1,0.1))
s.label(newdata,clab=0,cpoi=0)
s.label(data.frame(G,V),pch=c(1,20)[gardon+1],cpoi=2,clab=0,add.p=T)
lev0 <- c(0.05,0.1,0.3,0.5,0.7,0.9)
contour(newG,newV,matrix(newresult,20),xlab="G",ylab="V",add=T,lwd=2,col="red",levels=lev0)
par(mar=old)
```



Des commentaires ?

8.4 Modèles complexes

Et l'altitude ?

```
A <- gardonmi$A
glm4 <- glm(gardon~V+I(V^2)+G+I(G^2)+A+I(A^2), family=binomial)
anova(glm4,test="Chisq")
```

```
Analysis of Deviance Table
Model: binomial, link: logit

Response: gardon

Terms added sequentially (first to last)
```

```
       Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                     644     787.17
V       1  135.122       643     652.04 < 2.2e-16
I(V^2)  1   16.756       642     635.29 4.251e-05
G       1  127.628       641     507.66 < 2.2e-16
I(G^2)  1   11.213       640     496.45 0.0008122
A       1   21.411       639     475.04 3.706e-06
I(A^2)  1   12.339       638     462.70 0.0004435
```

Et le bassin ?

```
S <- gardonmi$S
glm5 <- glm(gardon~V+I(V^2)+G+I(G^2)+A+I(A^2)+S, family=binomial)
anova(glm5,test="Chisq")
```

```
Analysis of Deviance Table
Model: binomial, link: logit

Response: gardon

Terms added sequentially (first to last)


       Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                     644     787.17
V       1  135.122       643     652.04 < 2.2e-16
I(V^2)  1   16.756       642     635.29 4.251e-05
G       1  127.628       641     507.66 < 2.2e-16
I(G^2)  1   11.213       640     496.45 0.0008122
A       1   21.411       639     475.04 3.706e-06
I(A^2)  1   12.339       638     462.70 0.0004435
S       7   22.969       631     439.73 0.0017256
```

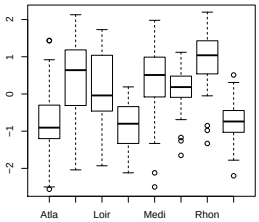
La difficulté :

```
S <- gardonmi$S
glm6 <- glm(gardon~V+I(V^2)+G+I(G^2)+S+A+I(A^2), family=binomial)
anova(glm6,test="Chisq")
```

```
Analysis of Deviance Table
Model: binomial, link: logit

Response: gardon

Terms added sequentially (first to last)


       Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                     644     787.17
V       1  135.122       643     652.04 < 2.2e-16
I(V^2)  1   16.756       642     635.29 4.251e-05
G       1  127.628       641     507.66 < 2.2e-16
I(G^2)  1   11.213       640     496.45 0.0008122
S       7   12.425       633     484.02 0.0874102
A       1   35.885       632     448.14 2.093e-09
I(A^2)  1    8.410       631     439.73 0.0037322
```

`S` est un effet significatif derrière `A` mais pas devant. Les deux variables sont liées :

```
plot(S,A)
```



Introduire l'une ou l'autre des variables donne -il les mêmes prédictions ? Le terme carré en `A` a-t-il autant d'effet que le terme en `S` ? Voir aussi **step** (choix d'un modèle pas à pas), **add1** (ajouter une variable dans un modèle), **drop1** (enlever une variable dans un modèle).
Et les interactions ? On dit que la statistique est un art...

9 Modèle poissonnien

On reprend l'exemple `ecrin` des fiche tdr14b et tdr33.

```
ecrin <- read.table("http://pbil.univ-lyon1.fr/R/donnees/ecrin.txt", h = TRUE)
ecrin[1:5,]
```

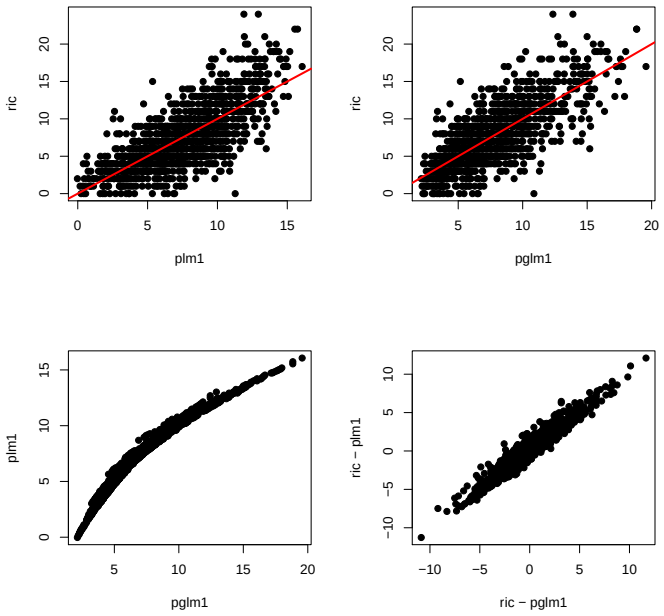
```
  STA SEM HEU RIC
1   3   2   1   5
2   3   2   2   3
3   3   3   1   5
4   3   3   2   3
5   3   4   1   4
```

```
ric <- ecrin$RIC
heu <- factor(ecrin$HEU)
levels(heu) <- c("matin","soir")
sem <- factor(ecrin$SEM)
sta <- factor(ecrin$STA)
lm1 <- lm(ric~heu+sem+sta+sem:sta)
anova(lm1)
```

```
Analysis of Variance Table
Response: ric
           Df Sum Sq Mean Sq  F value    Pr(>F)
heu         1 3071.3 3071.28 561.3111 < 2.2e-16
sem        51 6133.2  120.26  21.9786 < 2.2e-16
sta        13 3518.8  270.68  49.4690 < 2.2e-16
sem:sta   622 6242.1   10.04   1.8341 2.698e-14
Residuals 627 3430.7    5.47
```

Est-il vraiment utile biologiquement d'estimer 622 paramètres supplémentaires ? La variable réponse est un effectif et son erreur intrinsèque est discrète. Si il y a réellement 6 espèces, on en trouvera 7 sur une erreur d'identification ou 4 si il se met à pleuvoir, mais rarement 5.5. Les modèles à erreur poissonienne sont faits pour ce type de données. Comparer les modèles sans interactions :

```
lm1 <- lm(ric~heu+sem+sta)
glm1 <- glm(ric~heu+sem+sta,family=poisson)
plm1 <- predict(lm1)
pglm1 <- predict(glm1,type="response")
par(mfrow=c(2,2))
plot(plm1,ric,pch=19)
abline(0,1,col="red",lwd=2)
plot(pglm1,ric,pch=19)
abline(0,1, col="red",lwd=2)
plot(pglm1,plm1,pch=19)
plot(ric-pglm1,ric-plm1,pch=19)
```



Les modifications sont donc sensibles. Le lien joue un rôle non négligeable. Mais les deux modèles sont de précision voisine, parce que l'erreur aléatoire dans ce modèle est considérable.

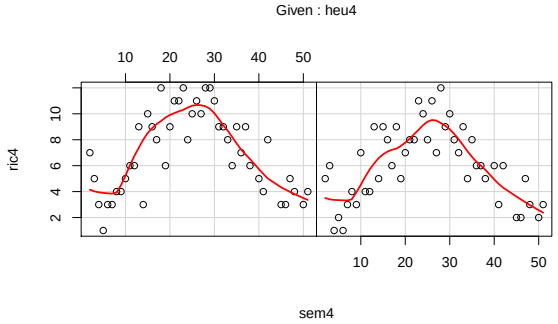
```
 sum((ric-plm1)^2)
[1] 9672.783
 sum((ric-pglm1)^2)
[1] 9154.22
```

9.1 Une étude partielle

Extraire ce qui concerne la station 4 :

```
 ecrin4 <- ecrin[ecrin$STA==4,]
 ric4 <- ecrin4$RIC
 heu4 <- as.factor(ecrin4$HEU)
 sem4 <- ecrin4$SEM
 coplot(ric4~sem4|heu4,show=F,panel=function (x,y,...)
 panel.smooth(x,y,span=0.3,...,lwd=2))
 lm4 <- lm(ric4~sem4+I(sem4^2)+I(sem4^3)+heu4)
 anova(lm4)
Analysis of Variance Table
Response: ric4
          Df Sum Sq Mean Sq  F value  Pr(>F)
sem4       1   0.02    0.02   0.0042 0.94849
I(sem4^2)  1 474.48  474.48 125.0079 < 2e-16
I(sem4^3)  1   0.18    0.18   0.0482 0.82668
heu4       1  22.01   22.01   5.7991 0.01815
Residuals 87 330.22    3.80
 glm4 <- glm(ric4~sem4+I(sem4^2)+I(sem4^3)+heu4,family=poisson)
 anova(glm4,test="Chisq")
Analysis of Deviance Table
Model: poisson, link: log

Response: ric4
```

```
Terms added sequentially (first to last)

          Df Deviance Resid. Df Resid. Dev Pr(>Chi)
NULL                         91    136.185
sem4       1    0.002        90    136.183  0.96061
I(sem4^2)  1   81.257        89     54.926  < 2e-16
I(sem4^3)  1    0.000        88     54.926  0.99582
heu4       1    3.373        87     51.554  0.06629
```



On introduit des différences sensibles. Continuer en étudiant d'autres stations.

9.2 Nombres d'accidents

On extrait de l'ouvrage de P. Eddy, E. Potter, & B. Page [1]p. 330-332 la date en nombre de jours entre le 01/01/1972 (x=1) et le 31/12/1975 (x=1461) de 142 catastrophes aériennes. Les données sont dans :

`http://pbil.univ-lyon1.fr/R/donnees/cata.txt`

1. Partager la période d'étude en unités de 73 jours et compter le nombre d'accidents par période. Éditer ces effectifs. Caractériser l'amélioration de la sécurité aérienne sur cette période par une régression sur le numéro d'ordre de la période dans le temps. L'erreur attendue sur les comptages est poissonienne. Comparer l'anova avec un test du χ^2. Prédire le nombre d'accident pour les périodes suivantes sous l'hypothèse de continuité du phénomène.

2. Pour les 138 premières catastrophes, compter le nombre de jours d'attente de la catastrophe suivante. Caractériser l'amélioration de la sécurité aérienne sur cette période par une régression sur la date. Dans un processus localement aléatoire le temps d'attente a une distribution exponentielle.

3. Montrer que les modèles sont voisins bien que les analyses soient différentes. Un détail les sépare cependant. Lequel ?

Références

[1] P. Eddy, E. Potter, and B. Page. *Destination dstre*. Grasset, Paris, 1976.

[2] T. Oberdorff, D. Pont, B. Hugueny, and D. Chessel. A probabilistic model characterizing fish assemblages of french rivers : a framework for the adaptation of a fish-based index. *Freshwater Biology*, 46 :399–415, 2001.

[3] C.J.F. Ter Braak and C.W.N. Looman. Weighted averaging, logistic regression and the gaussian response model. *Vegetatio*, 65 :3–11, 1986.