

Modèles Linéaires Appliqués / Régression GLM : Lien & Variance

Arthur Charpentier

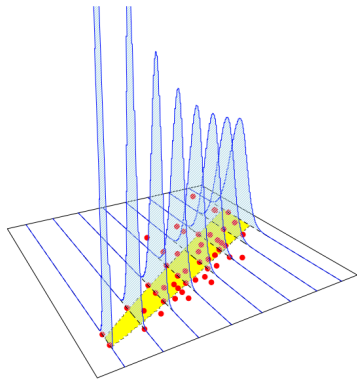
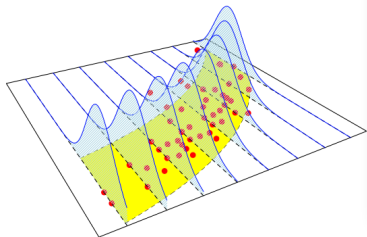
UQAM

Hiver 2020 - COVID-19 # 14

GLM, Lien & Variance

Les GLM sont associés à deux composantes importantes

- lien : $\mathbb{E}(Y) = \mu = g^{-1}(\eta) = g^{-1}(\mathbf{x}^\top \boldsymbol{\beta})$
- variance : $\text{Var}(Y) = \varphi V(\mu)$



Quasi-Vraisemblance

La plupart des caractéristiques des GLM ne dépendent *que* des deux premiers moments de la distribution

- lien : $\mathbb{E}(Y) = \mu = g^{-1}(\eta) = g^{-1}(\mathbf{x}^\top \boldsymbol{\beta})$
- variance : $\text{Var}(Y) = \varphi V(\mu)$

Quasi log-densité :

$$Q(y_i; \mu_i) = \int_{y_i}^{\mu_i} \frac{y_i - t}{\varphi V(t)} dt$$

Quasi log-vraisemblance :

$$Q(\boldsymbol{\mu}, \varphi, \mathbf{y}) = \sum_{i=1}^n Q(\mu_i, y_i) = \sum_{i=1}^n \int_{y_i}^{\mu_i} \frac{y - t}{\varphi V(t)} dt$$

Quasi score :

$$S_j(\beta_j) = \frac{\partial}{\partial \beta_j} Q(\boldsymbol{\beta}, \varphi, \mathbf{y}) = \sum_{i=1}^n \frac{\partial \mu_i}{\partial \beta_j} \frac{y_i - \mu_i(\beta_j)}{\varphi V(\mu_i)}$$

Quasi-Vraisemblance

Cas assez général, transformations puissances

- lien : $\eta = \mu^a$ ou $\mathbb{E}(Y) = \mu = \eta^{1/a} = (\mathbf{x}^\top \boldsymbol{\beta})^{1/a}$
- variance : $\text{Var}(Y) = \varphi \mu^b$

```
1 > glm(y~x,family=tweedie(var.power=b,link.power=a))
```

On peut définir une loi **quasi-Poisson**

Poisson $\rightarrow \mathbb{E}(Y|\mathbf{X} = \mathbf{x}) = \exp(\mathbf{x}^\top \boldsymbol{\beta}) = \mu$ et $\text{Var}(Y|\mathbf{X} = \mathbf{x}) = \mu$

$\mathbb{E}(Y|\mathbf{X} = \mathbf{x}) = \exp(\mathbf{x}^\top \boldsymbol{\beta})$ et $\text{Var}(Y|\mathbf{X} = \mathbf{x}) = \varphi \mu \rightarrow$ **quasi-Poisson**

On peut aussi définir une loi **quasi-Binomiale**,

$$\mu = \frac{e^{\mathbf{x}^\top \boldsymbol{\beta}}}{1 + e^{\mathbf{x}^\top \boldsymbol{\beta}}} \text{ et } V(\mu) = \varphi \mu(1 - \mu)$$

Quasi-Poisson

Loi binomiale négative, de moyenne $\mathbb{E}(Y) = \frac{n(1-p)}{p}$

$$f(y, n, p) = \frac{\Gamma(y+n)}{\Gamma(n)y!} p^n (1-p)^y, \quad y \in \mathbb{N},$$

```
1 > Y = rnbinom(n=1e5, size=10, prob=.1)
```

ici $\mathbb{E}(Y) = 90$ (et $\log(90) = 4.49981$)

```
2 > summary(glm(Y~1, family=quasipoisson))
```

```
3
```

```
4 Coefficients:
```

```
5             Estimate Std. Error t value Pr(>|t|)
```

```
6 (Intercept) 4.499805    0.001055    4266   <2e-16 ***
```

```
7
```

```
8 (Dispersion parameter quasipoisson family 10.01436)
```

Quasi-Poisson

```
1 > glm(formula = Y ~ YEARMARRIAGE + CHILDREN +  
    RELIGIOUS + EDUCATION + SATISFACTION, family =  
    poisson, data = base)
```

2

3 Coefficients:

	Estimate	Std. Error	z value	Pr(>z)	
4 (Intercept)	-0.83704	0.44538	-1.879	0.06019	.
5 YEARMARRIAGE	0.02287	0.01189	1.923	0.05452	.
6 CHILDREN	0.52029	0.16889	3.081	0.00207	**
7 RELIGIOUS	-0.24972	0.04802	-5.201	1.99e-07	***
8 EDUCATION	0.10974	0.02288	4.797	1.61e-06	***
9 SATISFACTION	-0.34343	0.04555	-7.540	4.72e-14	***

10

11

12 (Dispersion parameter poisson family taken to 1)

Quasi-Poisson

```
1 > glm(formula = Y ~ YEARMARRIAGE + CHILDREN +  
    RELIGIOUS + EDUCATION + SATISFACTION, family =  
    quasipoisson, data = base)
```

2

3 Coefficients:

	Estimate	Std. Error	t value	Pr(>t)	
4 (Intercept)	-0.83704	0.84741	-0.988	0.32370	
5 YEARMARRIAGE	0.02287	0.02263	1.011	0.31267	
6 CHILDREN	0.52029	0.32134	1.619	0.10599	
7 RELIGIOUS	-0.24972	0.09136	-2.733	0.00647	**
8 EDUCATION	0.10974	0.04353	2.521	0.01198	*
9 SATISFACTION	-0.34343	0.08667	-3.963	8.38e-05	***

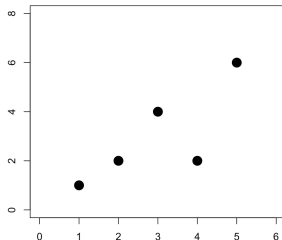
10

11

12 (Dispersion parameter for quasipoisson taken to 3.620)

Jeu de Données

```
1 > x = c(1,2,3,4,5)
2 > y = c(1,2,4,2,6)
3 > plot(x,y,pch=19)
4 > base = data.frame(x=x, y=y)
```

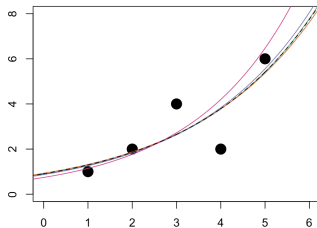
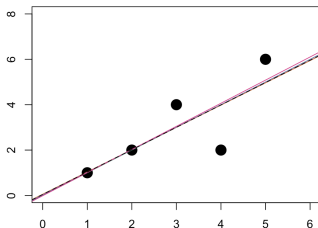


On considère plusieurs modèles de régression

```
1 > regNId = glm(y~x,family=gaussian(link="identity"))
2 > regNlog = glm(y~x,family=gaussian(link="log"))
3 > regPIId = glm(y~x,family=poisson(link="identity"))
4 > regPlog = glm(y~x,family=poisson(link="log"))
5 > regGId = glm(y~x,family=Gamma(link="identity"))
6 > regGlog = glm(y~x,family=Gamma(link="log"))
7 > regIGId = glm(y~x,family=inverse.gaussian(link= ...
8 > regIGlog = glm(y~x,family=inverse.gaussian(link= ..
```

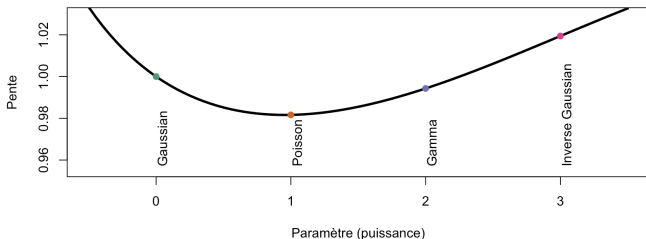

Jeu de Données

```
1 > DF=data.frame(NId=predict(regNId,type="response"),
2 +               Nlog=predict(regNlog,type="response"),
3
4 > DF
5   obs  NId  Nlog  PInd  Plog  GId  Glog  TwId  Twlog
6 1    1    1 1.277 1.037 1.319 1.021 1.261 1.031 1.299
7 2    2    2 1.833 2.018 1.874 2.016 1.827 2.016 1.857
8 3    4    3 2.631 3.000 2.661 3.010 2.647 3.002 2.653
9 4    2    4 3.776 3.982 3.779 4.004 3.835 3.987 3.791
10 5    6    5 5.419 4.963 5.367 4.999 5.557 4.973 5.418
```



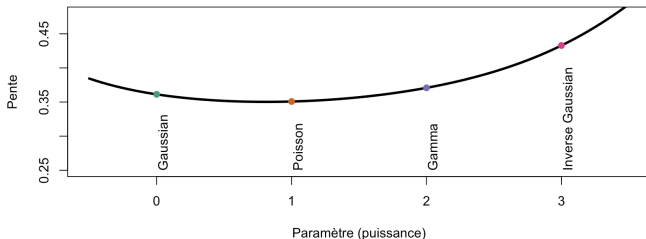
Pentes des modèles avec différentes lois, avec un lien 'identité'

```
1 > pente = function(gamma) summary(glm(y~x,family=
  tweedie(var.power=gamma,link.power=1),data=base))
  $coefficients[2,1:2]
```



Pentes des modèles avec différentes lois, avec un lien 'log'

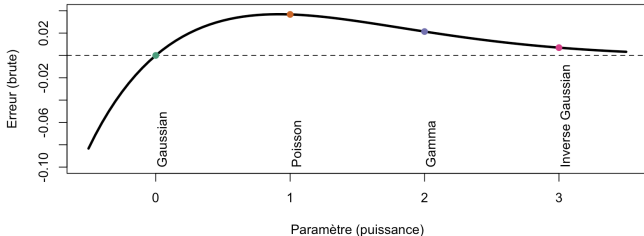
```
1 > pente = function(gamma) summary(glm(y~x,family=
  tweedie(var.power=gamma,link.power=0),data=base))
  $coefficients[2,1:2]
```



Jeu de Données

Erreur de prévision $y_1 - \hat{\mu}_1$ avec différentes lois,
avec un lien 'identité'

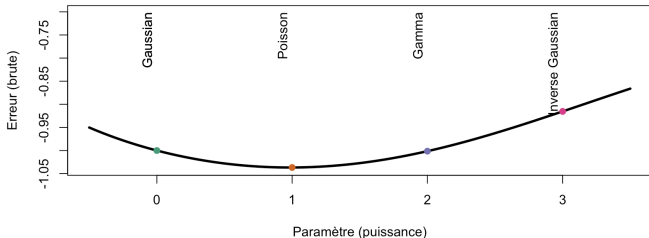
```
1 > pente = function(gamma) summary(glm(y~x,family=
    tweedie(var.power=gamma,link.power=0),data=base))
    $coefficients[2,1:2]
```



Jeu de Données

Erreur de prévision $y_5 - \hat{\mu}_5$ avec différentes lois,
avec un lien 'identité'

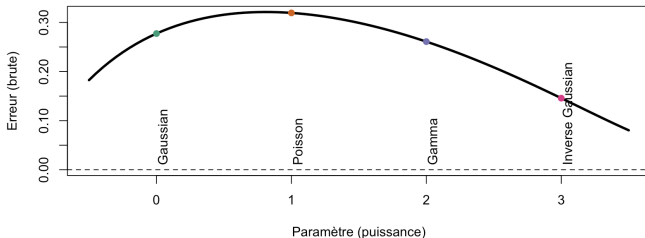
```
1 > pente = function(gamma) summary(glm(y~x,family=  
  tweedie(var.power=gamma,link.power=0),data=base))  
  $coefficients[2,1:2]
```



Jeu de Données

Erreur de prévision $y_1 - \hat{\mu}_1$ avec différentes lois,
avec un lien 'log'

```
1 > pente = function(gamma) summary(glm(y~x,family=
    tweedie(var.power=gamma,link.power=0),data=base))
    $coefficients[2,1:2]
```



Jeu de Données

Erreur de prévision $y_5 - \hat{\mu}_5$ avec différentes lois,
avec un lien 'log'

```
1 > pente = function(gamma) summary(glm(y~x,family=
    tweedie(var.power=gamma,link.power=0),data=base))
    $coefficients[2,1:2]
```

