Mars 2015

Modèle Linéaire Généralisé

Magali San Cristobal
INRA Toulouse – GenPhySE
magali.san-

Formation « Analyse statistique de données postgénomiques haut débit.
Module 3 : Statistiques avancées

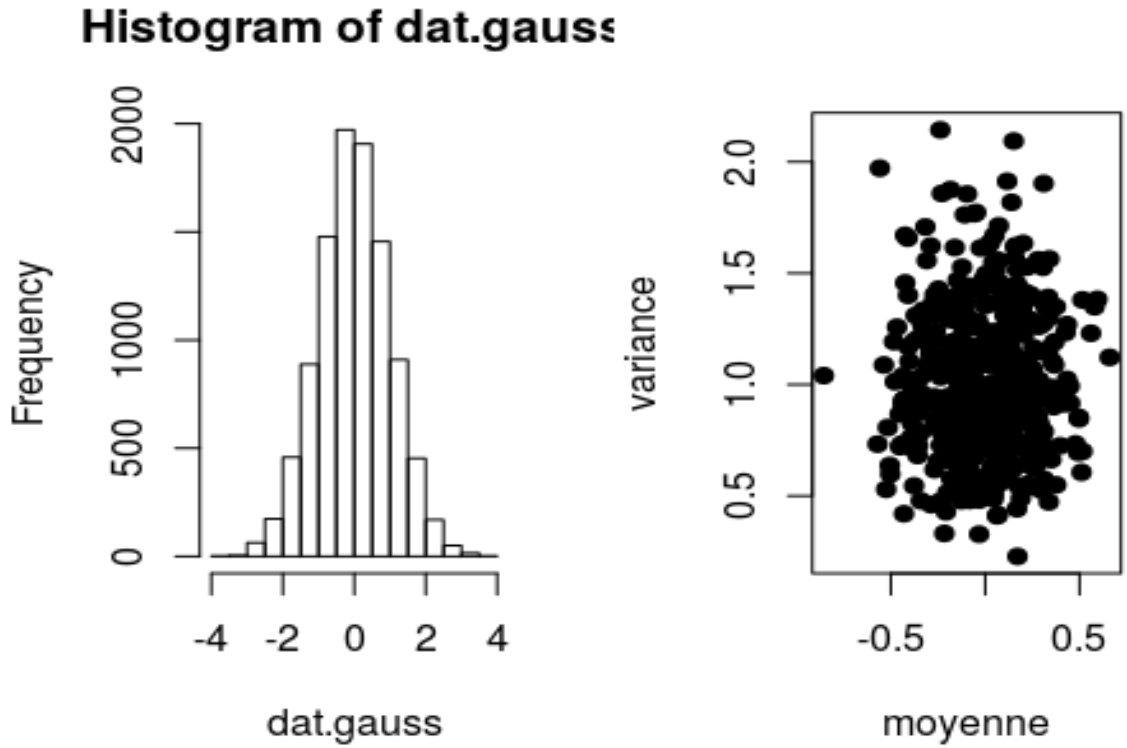
Introduction

- Contexte de la post-génomique
- Exemple : données transcriptomiques
- Statistique : Individus en ligne, variables en colonne
- Ici, gènes rangés en ligne, échantillons en colonne
- Analyse différentielle : un gène après l'autre
- Progression pas à pas basée sur les exemples concrets à l'aide du logiciel R

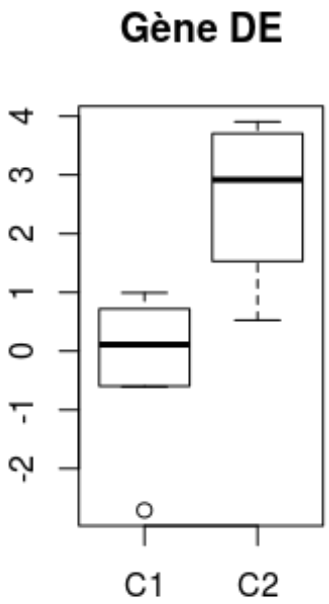
Rappels sur le modèle linéaire

	Condition 1			Condition 2		
	C1.1	…	C1.10	C2.1	…	C2.10
G1	0.01		-0.78	-1.47		1.82
G2	-0.98		-0.39	0.81		-0.40
…						
G500	-0.46		-2.7	0.22		1.43
Moyenne	0.17		-0.20	-0.41		0.53
Variance	0.67		0.89	0.95		1.10

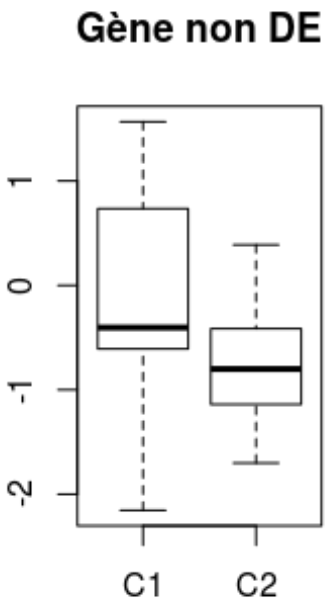
Courbe en cloche et pas de liaison moyenne-variance



Gènes différentiellement exprimés



N(0,1) vs N(2,1)



N(0,1) vs N(0,1)

Test de Student pour le gène G1 entre les 2 conditions du facteur :

```
t.test(dat.gauss["G1",] ~ facteur, var.equal=TRUE)
##
##  Two Sample t-test
##
## data:  dat.gauss["G1", ] by facteur
## t = -5.3635, df = 18, p-value = 4.256e-05
## alternative hypothesis: true difference in means
is not equal to 0
## 95 percent confidence interval:
##  -3.813931 -1.667009
## sample estimates:
## mean in group C1 mean in group C2
##       -0.0752177        2.6652520
```

```r
mod.gauss = lm ( dat.gauss["G1",] ~ facteur )
summary( mod.gauss )
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.07522    0.36129  -0.208    0.837
## facteurC2    2.74047    0.51095   5.364 4.26e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
' 1
##
## Residual standard error: 1.143 on 18 degrees of freedom
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.07522    0.36129  -0.208    0.837
## facteurC2    2.74047    0.51095   5.364 4.26e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
' 1
```

Exactement la même chose de le test de Student direct.

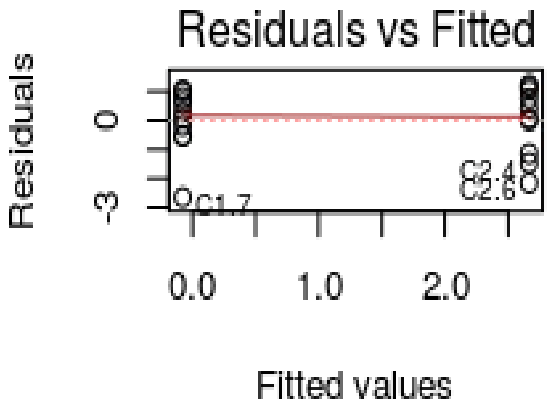
Autre possibilité : **le test du rapport de vraisemblance (LRT)**

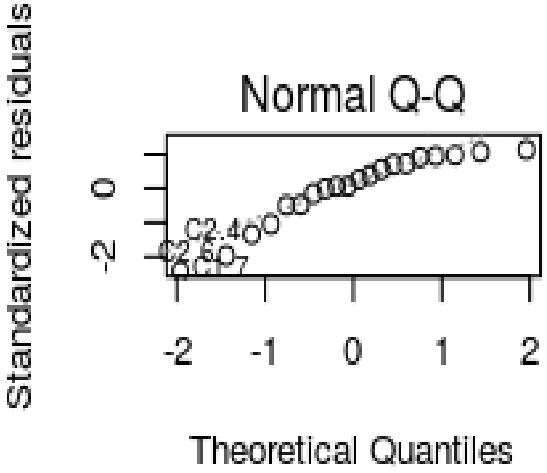
```
anova(mod.gauss, mod.gauss.reduit, test="LRT")
```

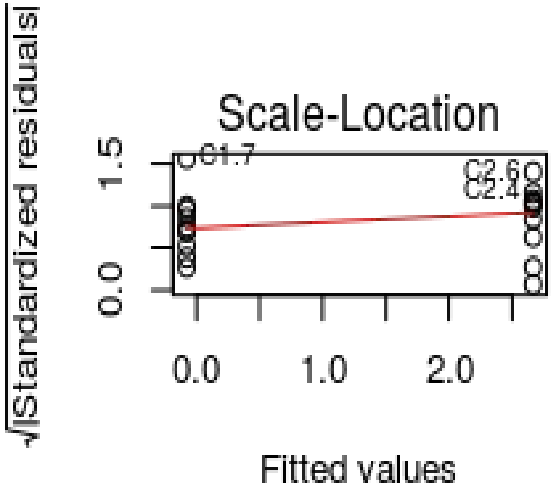
```
## Analysis of Variance Table
##
## Model 1: dat.gauss["G1", ] ~ facteur
## Model 2: dat.gauss["G1", ] ~ 1
##   Res.Df    RSS Df Sum of Sq  Pr(>Chi)
## 1     18 23.496
## 2     19 61.047 -1   -37.551 8.162e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
' 1
```

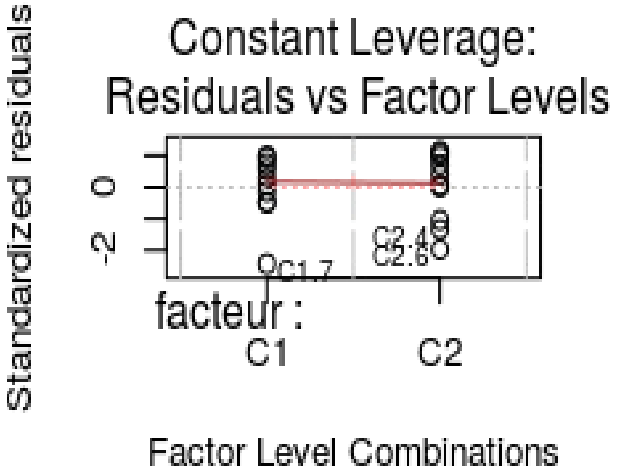
Vérification des hypothèses :

résidus indépendants,
gaussiens et
de même loi (notamment variance (homoscédasticité))









On découvre une nouvelle fonction : **glm**

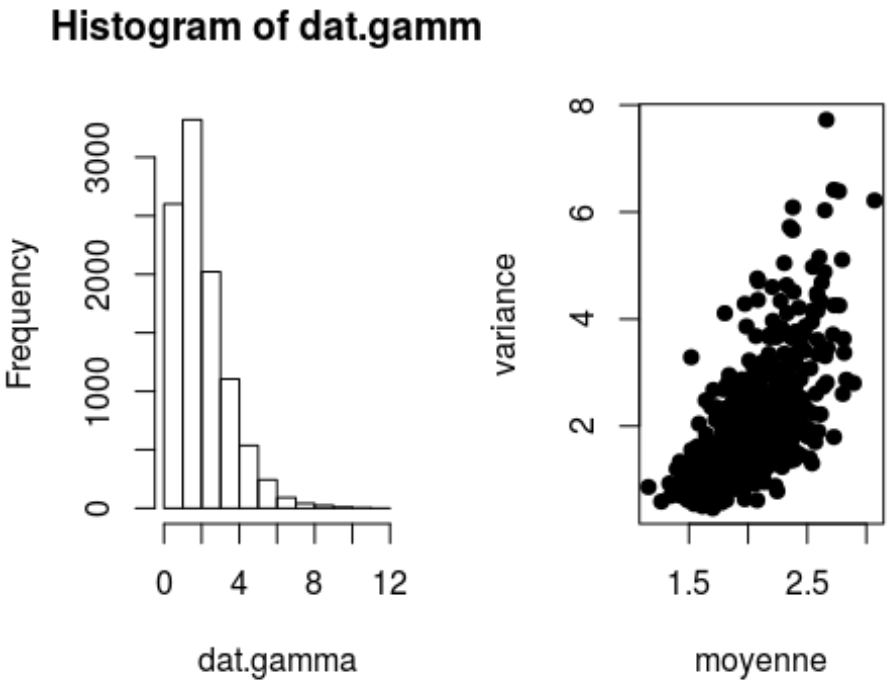
```
modg.gauss = glm ( dat.gauss["G1",] ~ facteur )
summary( modg.gauss )
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.07522    0.36129  -0.208    0.837
## facteurC2    2.74047    0.51095   5.364 4.26e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
' 1
##
## (Dispersion parameter for gaussian family taken to be
1.305337)
##
```

```
## Residual deviance: 23.496  on 18  degrees of freedom
```

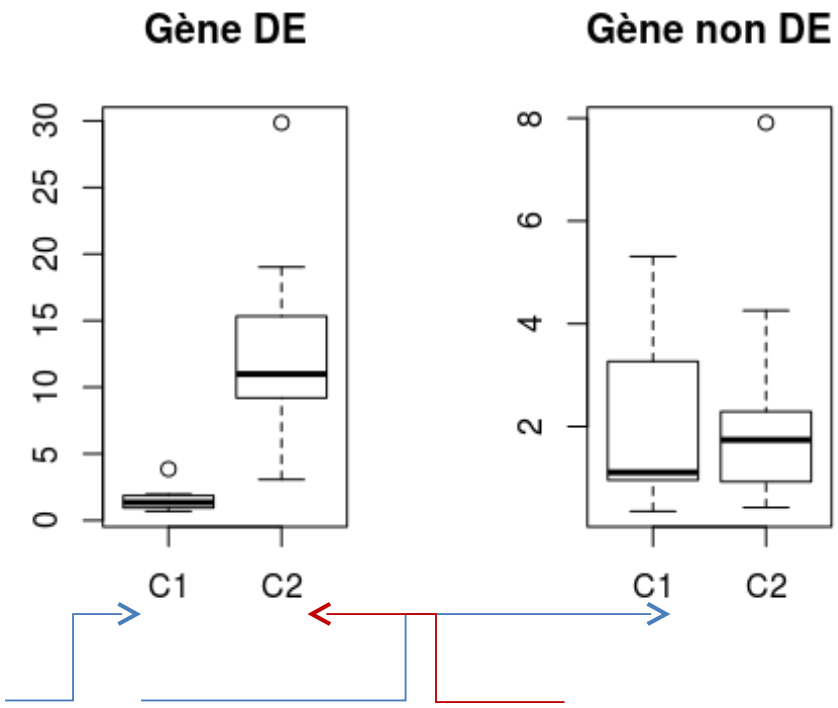
D*

Données Gamma

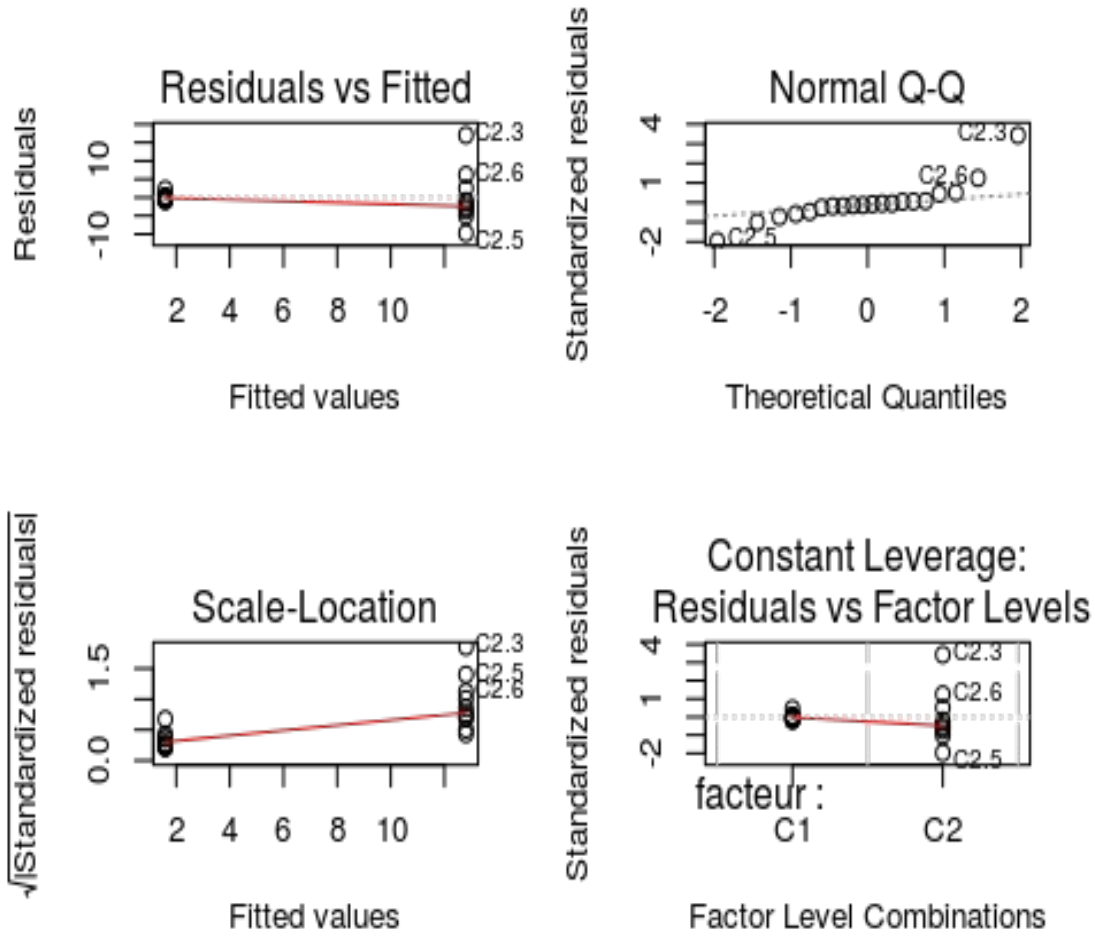


Distribution non symétrique et relation moyenne - variance

```r
a = 2;s = 1
dat.gamma = rgamma(10000, shape=a, scale=s)
```


Gène DE
Gène non DE
0
5
10
15
20
25
30
2
4
6
8
C1
C2
C1
C2

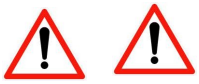
Si on ajuste un modèle linéaire, on a des problèmes :



Non normalité, mais surtout hétéroscédasticité

Notre premier « vrai » **modèle linéaire généralisé** :

```
mod.gamma = glm ( dat.gamma["G1",] ~ facteur,
                  family=Gamma(link="inverse")  )

## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)

## (Intercept)   0.6275     0.1131   5.549 2.88e-05
***
## facteurC2    -0.5494     0.1140  -4.821 0.000137
***
```



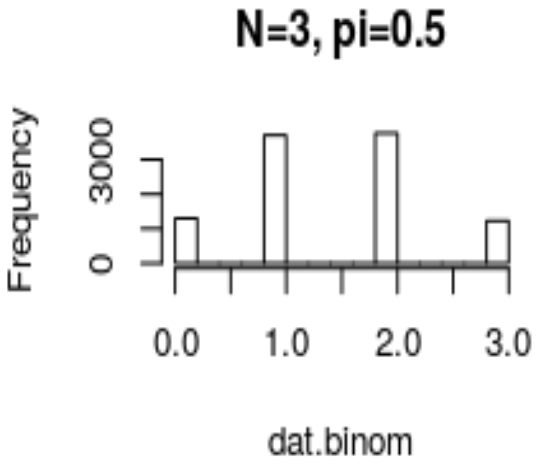
```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.6275     0.1131   5.549 2.88e-05 ***
## facteurC2    -0.5494     0.1140  -4.821 0.000137 ***
```

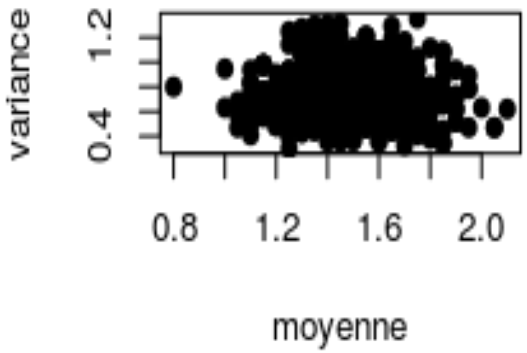
```
mod.gamma.reduit = glm ( dat.gamma["G1",] ~ 1,
family=Gamma(link="inverse")  )

anova( mod.gamma, mod.gamma.reduit, test="LRT")

## Analysis of Deviance Table
##
## Model 1: dat.gamma["G1", ] ~ facteur
## Model 2: dat.gamma["G1", ] ~ 1
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        18     5.2536
## 2        19    23.8930 -1  -18.639 3.57e-14 ***
```

Données binomiales

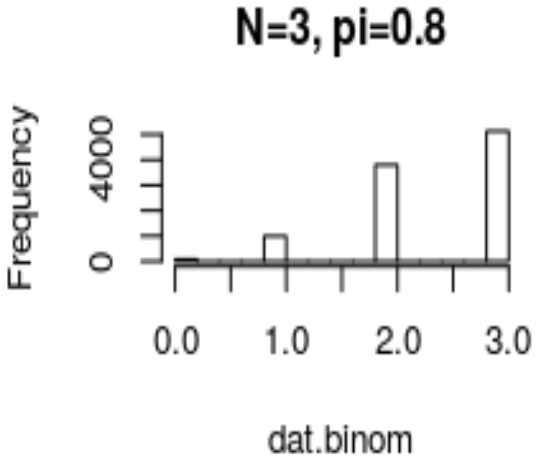
G1 2 1 1 3 0 2 0 1 2 0 2 2 0 2 2 1 0 3 2 1

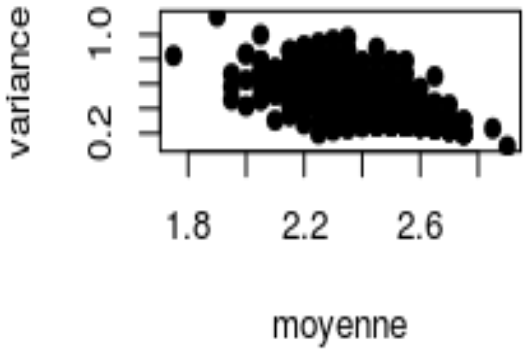
3 lancers pour chaque pièce





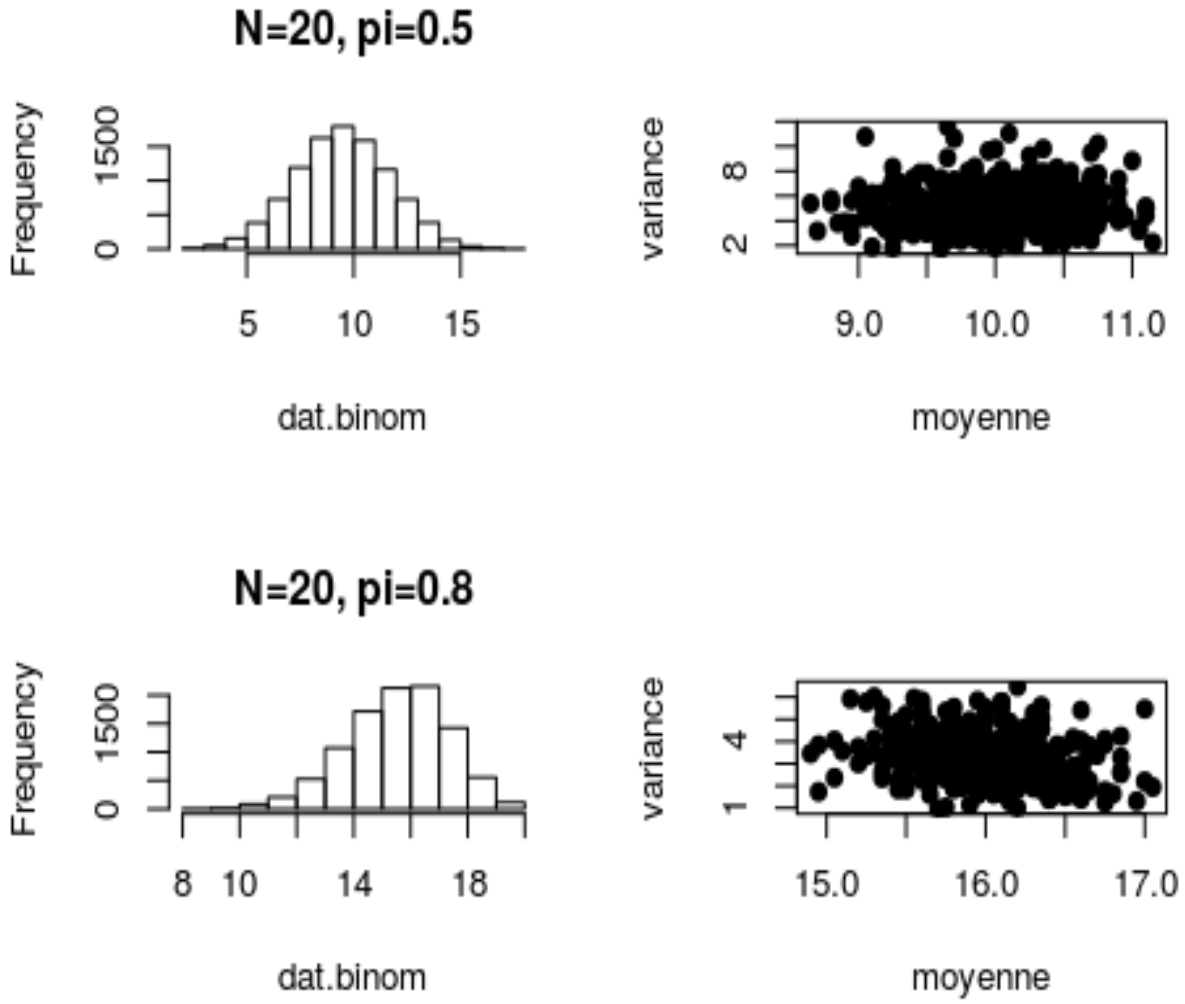
Pièce non pipée :
Distribution symétrique
Relation moyenne-variance « symétrique »



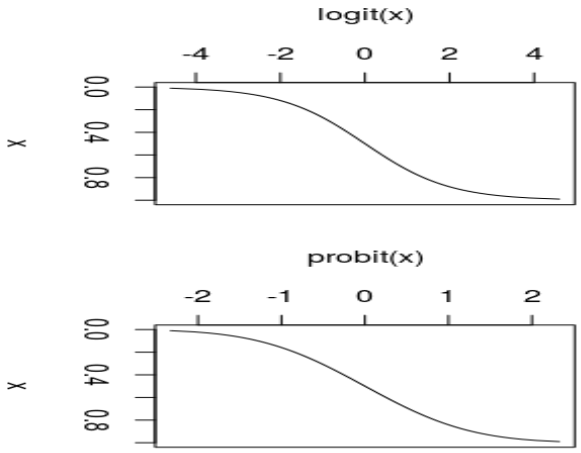


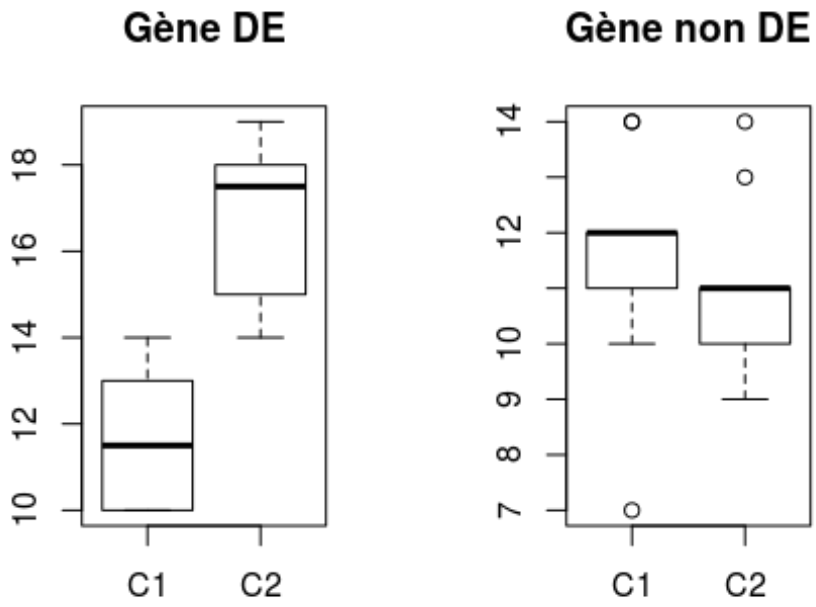
Pièce pipée :
Distribution asymétrique
Relation moyenne-variance

20 lancers pour chaque pièce



Théorème central limite


logit(x)
-4
-2
0
2
4
0.0
0.4
0.8
x
probit(x)
-2
-1
0
1
2
0.0
0.4
0.8
x



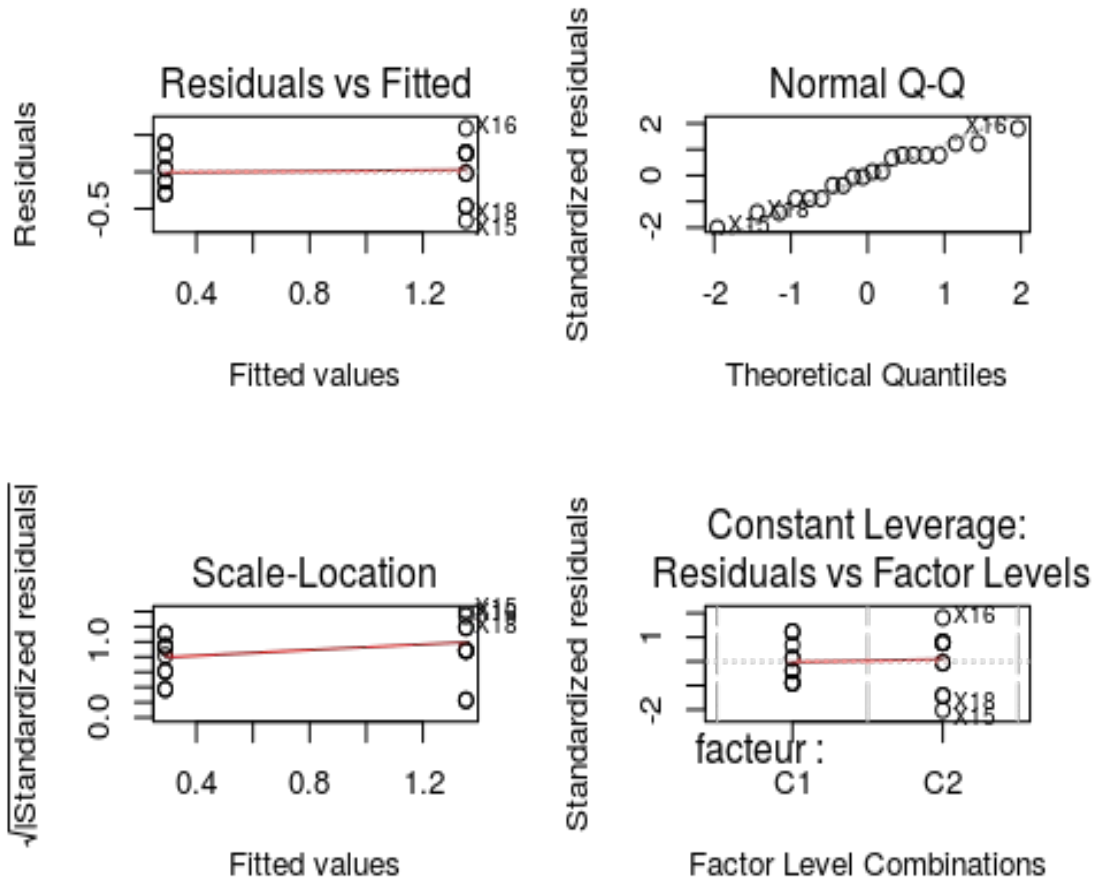
B(20, 0.6) B(20,0.8)

Je n’en ai pas parlé jusqu’ici, mais on peut très bien essayer un modèle linéaire sur données transformées :

```
p = dat.binom["G1",]/N
modlin.logitp = lm( logite(p) ~ facteur)
summary(modlin.logitp)
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.2904     0.1092   2.660   0.0159 *
## facteurC2     1.0617     0.1544   6.877 1.97e-06 ***
```

Et sur ces données-là, effectivement, ce n’est pas trop trop mal :



Mais il y a plus propre : le **modèle linéaire généralisé** !

```
mod.binom = glm ( p ~ facteur, weights=rep(N,20),
        family=binomial(link="logit") )

## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.3433     0.1435   2.392   0.0167 *
## facteurC2     1.3526     0.2424   5.579 2.41e-08 ***
```

```
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.3433     0.1435   2.392   0.0167 *
## facteurC2     1.3526     0.2424   5.579 2.41e-08 ***
```

```
mod.binom.reduit = glm ( p ~ 1, weights=rep(N,20), family=binomial(link="logit") )
anova(mod.binom, mod.binom.reduit, test="LRT")
```

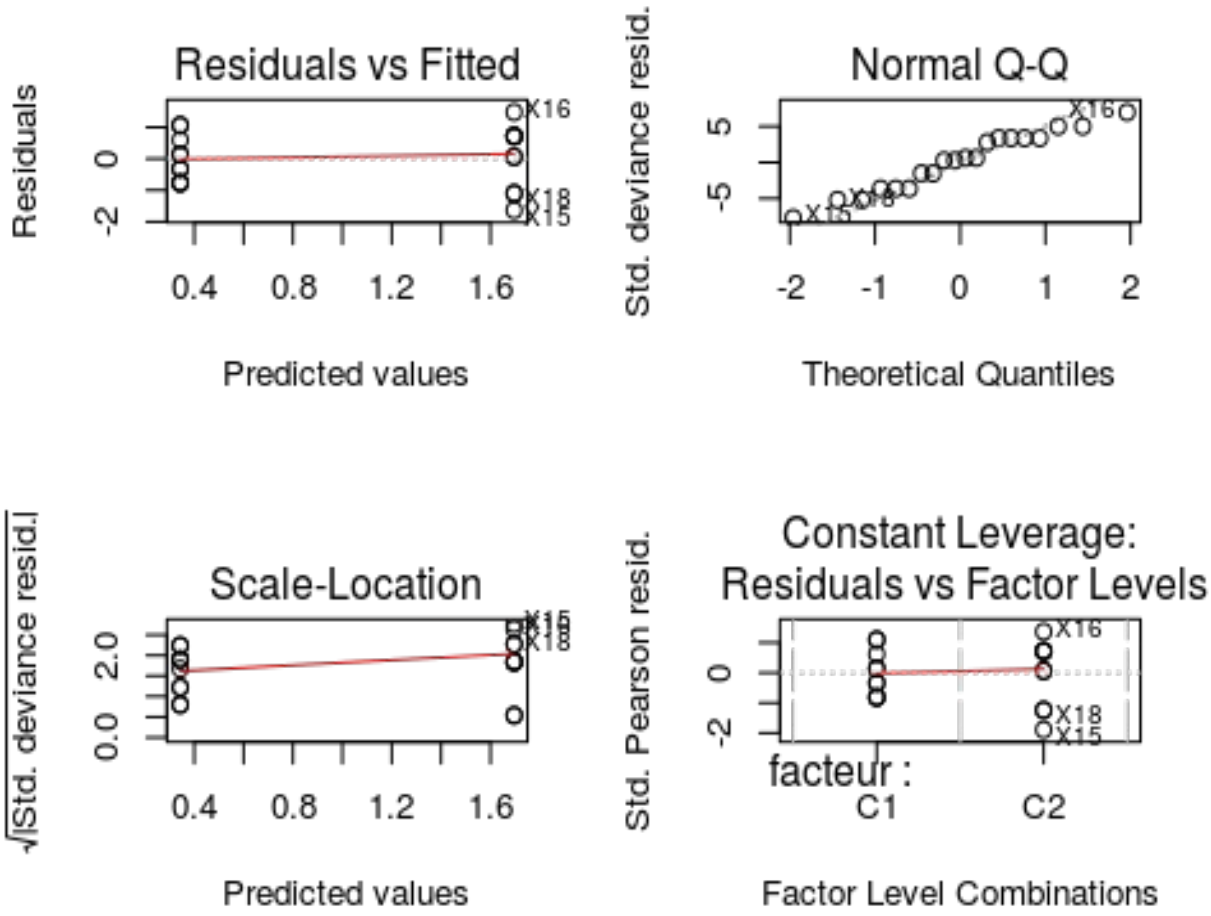
```
## Analysis of Deviance Table
##
## Model 1: p ~ facteur
## Model 2: p ~ 1
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1        18     13.936
## 2        19     48.062 -1  -34.126 5.165e-09 ***
```

Une petite idée de la théorie

Tests d'hypothèse

- test de **Wald**
- test "LRT" du **rapport de vraisemblance**

Divers types de **résidus**



Conclusion

Le modèle linéaire généralisé englobe plusieurs modèles connus :
* le modèle linéaire gaussien (avec l'ANOVA, ANCOVA, la régression, etc ...),
* la régression logistique (données binomiales),
* le modèle log-linéaire (données Poisson).

Il est applicable à toutes les lois de la **famille exponentielle**, dont : normale, Poisson, binomiale, gamma, gaussienne inverse.

Il permet d'obtenir des **estimations ponctuelles** des paramètres, des **estimations par intervalles**, et d'effectuer des **tests d'hypothèses**.

Quelques références

Ballesteros S. (2008)
http://rug.mnhn.fr/semin-r/PDF/semin-R_glm_SBallesteros_100608.pdf

CNAM (2007).
http://maths.cnam.fr/IMG/pdf/Presentation_MODGEN_02_2007.pdf

McCullagh P. et Nelder J.A. (1999). Generalized Linear Models. 2nd edition. Chapman & Hall.

http://wikistat.fr/, Le modèle linéaire général