Fiche d'utilisation du logiciel 

4-Modèle linéaire

D. Chessel & J. Thioulouse

Résumé

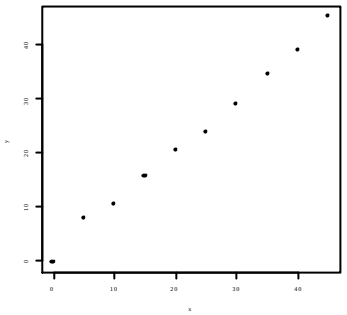
La fiche contient le matériel nécessaire pour des séances de travaux dirigés consacrées au modèle linéaire. Elle illustre en particulier la régression simple, la régression multiple, l'analyse de la variance et de la covariance.

Plan

1. Régression simple

Implanter le premier exemple proposé par Tomassone R., Charles-Bajard S. & Bellanger L. (1998) Introduction à la planification expérimentale, DEA « Analyse et modélisation des systèmes biologiques »:

```
> y_c(-0.6,7.9,10.5,15.4,20.3,23.8,28.8,34.7,39.1,45.4)
> x<-seq(from=0,to=45,by=5)
> x
 [1]  0  5 10 15 20 25 30 35 40 45
> y
 [1] -0.6  7.9 10.5 15.4 20.3 23.8 28.8 34.7 39.1 45.4

> plot(x,y)
```



lm

```
> ?lm

lm                    package:base                    R Documentation

Fitting Linear Models

Description:

     `lm' is used to fit linear models. It can be used to carry out
     regression, single stratum analysis of variance and analysis of
     covariance (although `aov' may provide a more convenient interface
     for these).

Usage:

     lm(formula, data, subset, weights, na.action,
        method = "qr", model = TRUE, x = FALSE, y = FALSE, qr = TRUE,
        singular.ok = TRUE contrasts = NULL, offset = NULL, ...)

     lm.fit (x, y,    offset = NULL, method = "qr", tol = 1e-7, ...)
     lm.wfit(x, y, w, offset = NULL, method = "qr", tol = 1e-7, ...)
     lm.fit.null (x, y,    method = "qr", tol = 1e-7, ...)
     lm.wfit.null(x, y, w, method = "qr", tol = 1e-7, ...)

> lm(y~x)

Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)            x
      0.791        0.966
```

1.1. Les objets « modèle linéaire »

Un modèle linéaire est un objet :

```
> lm1<-lm(y~x)
> lm1

Call:
lm(formula = y ~ x)
Coefficients:
(Intercept)            x
      0.791        0.966
```

lm1 est de la classe lm

```
> class(lm1)
[1] "lm"
```

La classe lm est une sous-classe de la classe list

```
> is.list(lm1)
[1] TRUE
```

lm1 est une collection de 12 composantes

```
> length(lm1)
[1] 12
> names(lm1)
 [1] "coefficients"  "residuals"     "effects"       "rank"
 [5] "fitted.values" "assign"        "qr"            "df.residual"
 [9] "xlevels"       "call"          "terms"         "model"
```

Noms et numéros des composantes de lm1

```
> lm1[[1]]
(Intercept)           x
     0.7909      0.9662

> lm1$coefficients
(Intercept)           x
     0.7909      0.9662

> lm1[[2]]
       1        2        3        4        5        6        7        8        9
-1.39091  2.27818  0.04727  0.11636  0.18545 -1.14545 -0.97636  0.09273 -0.33818
      10
 1.13091

>  lm1$residuals
       1        2        3        4        5        6        7        8        9
-1.39091  2.27818  0.04727  0.11636  0.18545 -1.14545 -0.97636  0.09273 -0.33818
      10
 1.13091
```

Le calcul est possible sur les composantes

```
> 2*lm1[[1]]
(Intercept)           x
      1.582       1.932
```

Fonctions génériques : summary

```
> summary(y)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   -0.6    11.7    22.1    22.5    33.2    45.4

> summary(lm1)
Call:
lm(formula = y ~ x)

Residuals:
   Min     1Q Median     3Q    Max
-1.391 -0.817  0.070  0.168  2.278

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.7909     0.6842    1.16     0.28
x             0.9662     0.0256   37.69  2.7e-10 ***
---
Signif. codes:  0  `***'  0.001  `**'  0.01  `*'  0.05  `.'  0.1  ` '  1

Residual standard error: 1.16 on 8 degrees of freedom
Multiple R-Squared: 0.994,      Adjusted R-squared: 0.994
F-statistic: 1.42e+003 on 1 and 8 degrees of freedom,   p-value: 2.69e-010
```

L'ordonnée à l'origine n'est pas significativement non nulle :

```
> lm2<-lm(y~-1+x)
> lm2

Call:
lm(formula = y ~ -1 + x)

Coefficients:
    x
0.991

> summary(lm2)

Call:
lm(formula = y ~ -1 + x)

Residuals:
   Min     1Q Median     3Q    Max
-0.979 -0.587  0.243  0.574  2.944

Coefficients:
  Estimate Std. Error t value Pr(>|t|)
x    0.991      0.014    70.6  1.2e-13 ***
---
Signif. codes:  0  `***'  0.001  `**'  0.01  `*'  0.05  `.'  0.1  ` '  1

Residual standard error: 1.19 on 9 degrees of freedom
Multiple R-Squared: 0.998,      Adjusted R-squared: 0.998
F-statistic: 4.98e+003 on 1 and 9 degrees of freedom,   p-value: 1.17e-013
```

Fonctions génériques : plot

```
> ?plot
```

plot package:base R Documentation

Generic X-Y Plotting

Description:

Generic function for plotting of R objects. For more details about the graphical parameter arguments, see `par'.

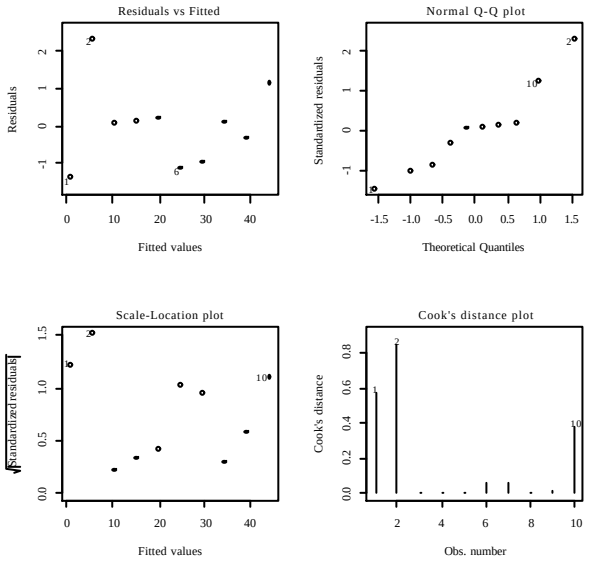
Usage:

```
     plot(x, ...)
     plot(x, y, xlim=range(x), ylim=range(y), type="p",
            main, xlab, ylab, ...)
     plot(y ~ x, ...)
```

```
> ?plot.lm
```

```
plot.lm                package:base                R Documentation

Plot Diagnostics for an lm Object

Description:

     Four plots (choosable by `which') are currently provided: a plot
     of residuals against fitted values, a Scale-Location plot of
     sqrt{| residuals |} against fitted values, a Normal Q-Q plot, and
     a plot of Cook's distances versus row labels.

Usage:

     plot.lm(x, which = 1:4,
            caption = c("Residuals vs Fitted", "Normal Q-Q plot",
                        "Scale-Location plot", "Cook's distance plot"),
            panel = points,
            sub.caption = deparse(x$call), main = "",
            ask = interactive() && one.fig && .Device != "postscript",
            ...,
            id.n = 3, labels.id = names(residuals(x)), cex.id = 0.25)
```

```
> plot(lm1) Que se passe t'il ?

> par(mfrow=c(2,2)) Pourquoi ?
> plot(lm1)
```



Graphique standard associé à un modèle linéaire

1) Résidus en fonction des valeurs prédites

2) Graphique quantile-quantile normal des résidus (normalité des résidus). N.B. Chacun des graphiques proposés est issu d'une recherche approfondie. Le qq-plot est de Wilk M.B. & Gnanadesikan R. (1968). Probability plotting methods for the analysis of data. Biometrika, *55, 1-17 validé par Cleveland W.S. (1994)* The elements of graphing data. *Hobart Press, Summit, New Jersey, p. 143. Les modes de lecture sont décrits dans des ouvrages célèbres comme Tuckey J.W.*

(1977) Exploratory data analysis, *Adsison-Wesley, Reading, Massachussets. Ici, les résidus sont sur- dispersés par rapport à une loi normale (cf. du Toit S.H.C., Steyn A.G.W. & Stumpf R.H. (1986)* Graphical Exploratory data analysis, *Springer-Verlag, , New-York, p. 49). Ouvrages classiques : Chambers J.M., Cleveland W.S., Kleiner B. & Tukey P.A. (1983)* Graphical methods for data analysis, *Wadsworth, Belmont, California. Cleveland W.S. (1993)* Visualizing data, *Hobart Press, Summit, New Jersey.*

3) Racine des valeurs absolues des résidus en fonction des valeurs prédites

4) Graphe des distances de Cook. Donne pour chacun des points de mesure la distance entre les paramètres estimés par la régression avec et sans ce point. Si l'importance du rôle de chaque point est concentré sur quelques valeurs, la régression n'est pas bonne (prise en compte de points aberrants). Voir Cook, R. D. and Weisberg, S. (1982). Residuals and Influence in Regression. *Chapman and Hall, New York.*

1.2. Normalité des résidus

On peut refaire l'expérience :

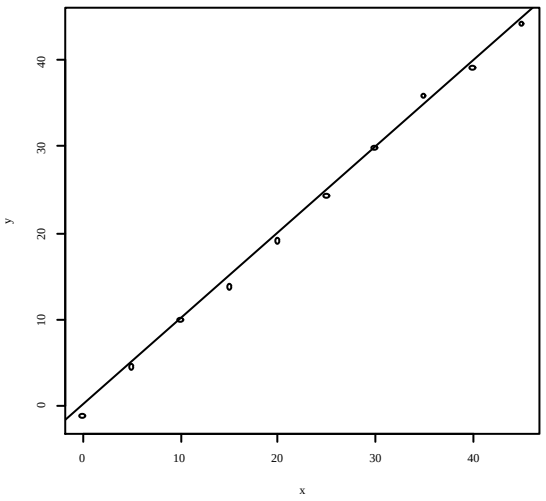
```
> x
 [1]  0  5 10 15 20 25 30 35 40 45
```

rnorm

```
> e<-rnorm(10)
> e
 [1] -1.4733 -0.7039 -0.2478 -1.4122 -0.9571 -0.8118 -0.2198  0.8083 -0.8420
[10] -0.7303
```

Calcul vectoriel

```
> y<-x+e
> y
 [1] -1.473  4.296  9.752 13.588 19.043 24.188 29.780 35.808 39.158 44.270

> par(mfrow=c(1,1)) (Sinon que se passe t'il ?)
> plot(x,y)
> abline(0,1)
```



abline

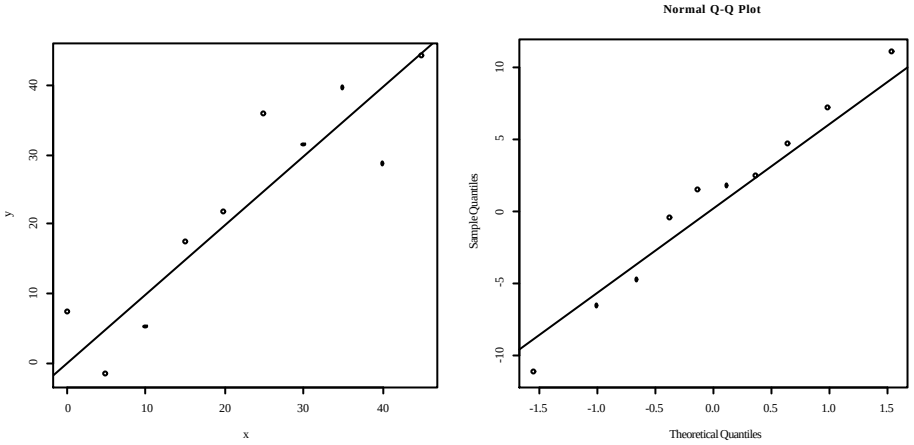
```
> ?abline
> abline(lm(y~x)) est-ce possible ?
> abline(lm(y~-1+x)) est-ce possible ?
```

Cet exercice est fondamental. Il construit les données conformément à un modèle. Une valeur de y est la réalisation d'une variable aléatoire gaussienne de moyenne m et de variance s^2. m est une fonction de x ($m = x$). On écrit $E(Y) = a * x$ (la moyenne est modélisée) et $Var(Y) = Cte = s^2$ (la variance est constante). L'erreur est gaussienne. Faire la régression, c'est estimer les valeurs inconnues (a, s) à partir de l'échantillon dans ce type de modèle (trop beau pour être « biologique » ?)

```
> summary(lm(y~-1+x))

Call:
lm(formula = y ~ -1 + x)

Residuals:
    Min      1Q  Median      3Q     Max
-1.4733 -0.6396 -0.3185 -0.0426  1.3558

Coefficients:
  Estimate Std. Error t value Pr(>|t|)
x   0.9844     0.0101    97.8  6.2e-15 ***
---
Signif. codes:  0  `***'  0.001  `**'  0.01  `*'  0.05  `.'  0.1  ` '  1
Residual standard error: 0.85 on 9 degrees of freedom
Multiple R-Squared: 0.999,      Adjusted R-squared: 0.999
F-statistic: 9.56e+003 on 1 and 9 degrees of freedom,   p-value: 6.22e-015
```

On a trouvé 0.9844 pour $a = 1$ et 0.85 pour $s = 1$.

```
> e<-rnorm(10,sd=5)
> y<-x+e
> lm3<-lm(y~-1+x)
> summary(lm3)

Call:
lm(formula = y ~ -1 + x)

Residuals:
   Min     1Q Median     3Q    Max
-11.31  -3.77   1.60   4.13  11.06

Coefficients:
  Estimate Std. Error t value Pr(>|t|)
x   0.9963     0.0794    12.6  5.2e-07 ***
---
Signif. codes:  0  `***'  0.001  `**'  0.01  `*'  0.05  `.'  0.1  ` '  1

Residual standard error: 6.7 on 9 degrees of freedom
Multiple R-Squared: 0.946,      Adjusted R-squared: 0.94
F-statistic:  158 on 1 and 9 degrees of freedom,        p-value: 5.24e-007

> plot(x,y)
> abline(lm3)
> qqnorm(lm3$residuals)
> qqline(lm3$residuals)
```



On a trouvé 0.9963 pour $a = 1$ et6.7 pour $s = 5$.

On peut reprendre le chapitre 9 « La régression linéaire simple » de Tomassone, R., Dervin, C. & Masson, J.P. (1993) Biométrie Modélisation de phénomènes biologiques. Masson, Paris. 1-553.

Exemple 1 (A faire en lisant le chapitre 9 « La régression simple » p. 175)

```
> lm(c(3,10)~-1+c(2,3))

Call:
lm(formula = c(3, 10) ~ -1 + c(2, 3))

Coefficients:
c(2, 3)
   2.77
> anova(lm(c(3,10)~-1+c(2,3)))
Analysis of Variance Table

Response: c(3, 10)
          Df Sum Sq Mean Sq F value Pr(>F)
c(2, 3)    1   99.7    99.7    10.7   0.19
Residuals  1    9.3     9.3
```

Exemple 2 (pour retrouver les détails des calculs décrits p.183)

```
> t_c(3,3,6,10,10,12,15,18,20)
> x_c(7,7,6,8,8,7,5,4,3)
> y_c(39.2,37.8,35.8,51.2,47.4,45.2,39.7,37.4,35.1)
```

Régression à deux variables sans terme constant :

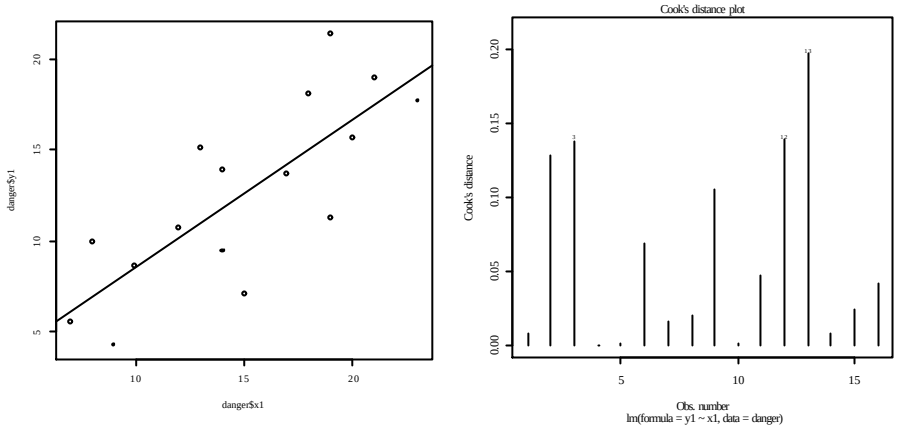
```
> lmt1_lm(y~-1+t+x)
> lmt1

Call:
lm(formula = y ~ -1 + t + x)

Coefficients:
    t      x
0.973  4.980  Estimation des coefficients

> predict(lmt1) Valeurs prédites
    1     2     3     4     5     6     7     8     9
37.78 37.78 35.72 49.57 49.57 46.53 39.49 37.43 34.40
> residuals(lmt1) Résidus
       1        2        3        4        5        6        7        8        9
```

```
 1.42277  0.02277  0.08357  1.63212 -2.16788 -1.33409  0.20646 -0.03274  0.70105

> summary(lmt1)

Call:
lm(formula = y ~ -1 + t + x)

Residuals:
    Min      1Q  Median      3Q     Max
-2.1679 -0.0327  0.0836  0.7010  1.6321

Coefficients:
  Estimate Std. Error t value Pr(>|t|)
t   0.9730     0.0541    18.0  4.1e-07 *** 0.973+2.36*0.0541=1.101
x   4.9798     0.1045    47.7  4.7e-10 ***
---
Signif. codes:  0  `***'  0.001  `**'  0.01  `*'  0.05  `.'  0.1  ` '  1

Residual standard error: 1.29 on 7 degrees of freedom
sqrt(1.673)=1.293 Estimation de la variance résiduelle
Multiple R-Squared: 0.999,      Adjusted R-squared: 0.999
F-statistic: 4.59e+003 on 2 and 7 degrees of freedom,   p-value: 1.22e-011

Décomposition de la variation p. 183
> sum(y*y)
[1] 15365
> sum(predict(lmt1)^2)
[1] 15354
> sum(residuals(lmt1)^2)
[1] 11.71

> anova(lmt1)
Response: y
          Df Sum Sq Mean Sq F value  Pr(>F)
t          1  11553   11553    6906 9.6e-12 ***
x          1   3800    3800    2272 4.7e-10 ***
Residuals  7     12       2
```

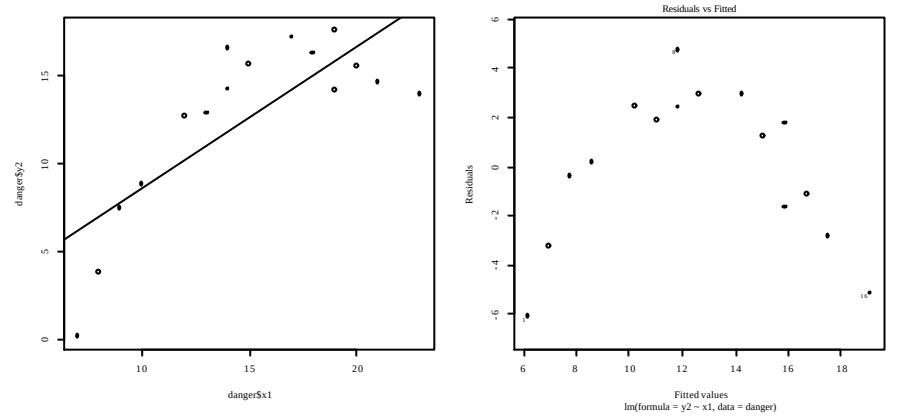
1.3. Les dangers de la régression simple

Utiliser les données du tableau danger.txt (tableau 1.1 dans Tomassone, R., Audrain, S., Lesquoy de Turckheim, E. & Millier, C. (1992) La régression. Masson, Paris. 1-188, p. 22).

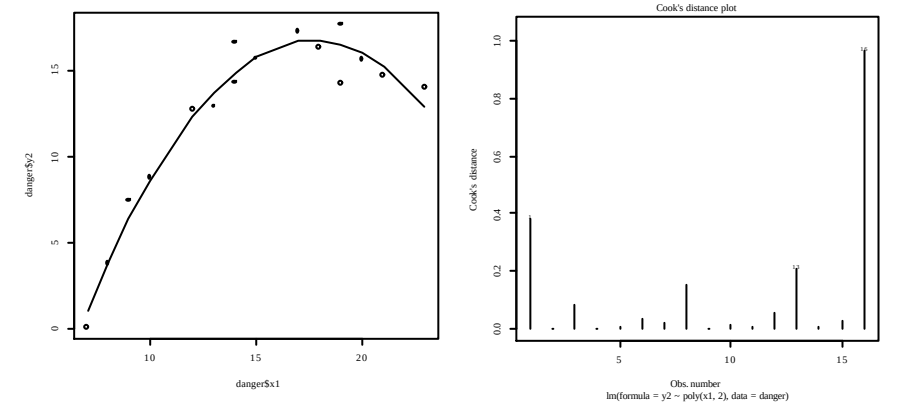
```
> danger
   x1     y1     y2     y3     y4    x2     y5
1   7  5.535  0.113  7.399  3.864 13.72  5.654
2   8  9.942  3.770  8.546  4.942 13.72  7.072
3   9  4.249  7.426  8.468  7.504 13.72  8.491
4  10  8.656  8.792  9.616  8.581 13.72  9.909
5  12 10.737 12.688 10.685 12.221 13.72  9.909
6  13 15.144 12.889 10.607  8.842 13.72  9.909
7  14 13.939 14.253 10.529  9.919 13.72 11.327
8  14  9.450 16.545 11.754 15.860 13.72 11.327
9  15  7.124 15.620 11.676 13.967 13.72 12.746
10 17 13.693 17.206 12.745 19.092 13.72 12.746
11 18 18.100 16.281 13.893 17.198 13.72 12.746
12 19 11.285 17.647 12.590 12.334 13.72 14.164
13 19 21.365 14.211 15.040 19.761 13.72 15.582
14 20 15.692 15.577 13.737 16.382 13.72 15.582
15 21 18.977 14.652 14.884 18.945 13.72 17.001
16 23 17.690 13.947 29.431 12.187 33.28 27.435

> plot(danger$x1,danger$y1)
> abline(lm1) Une bonne régression
> plot(lm1)
```



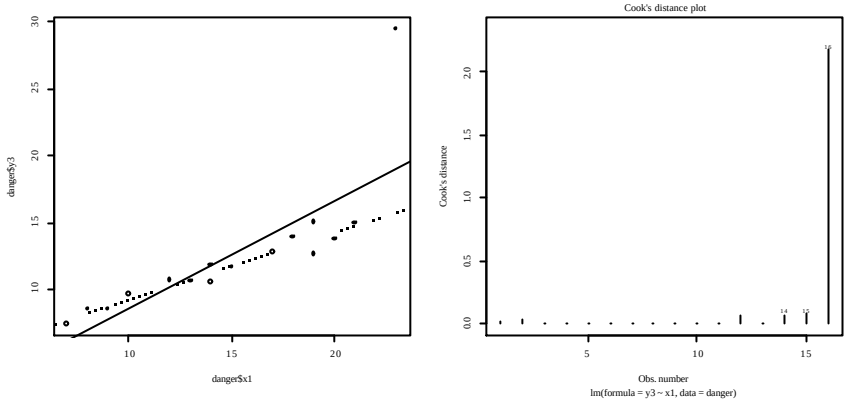

```
> lm2_lm(y2~x1,data=danger) Résidus autocorrélés
```



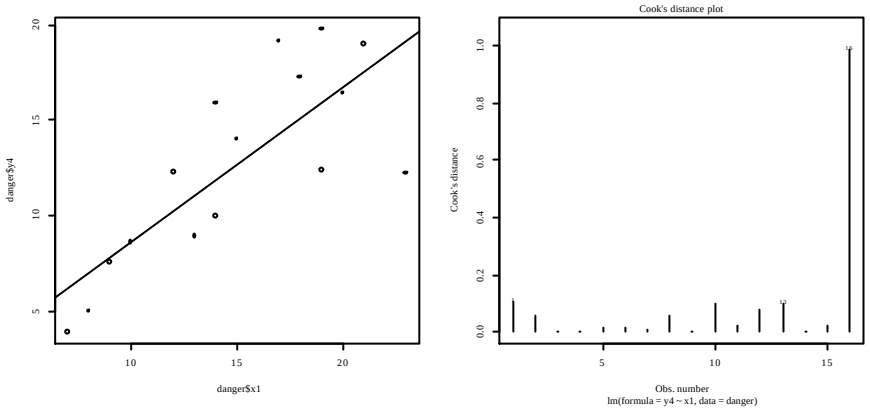

```
> lm2po_lm(y2~poly(x1,2),data=danger)
> plot(danger$x1,danger$y2)
> lines(danger$x1,predict(lm2po))
```




```
> lm3_lm(y3~x1,data=danger) Point aberrant
> abline(lm(danger$y3[1:15]~danger$x1[1:15]),lty=2)
```

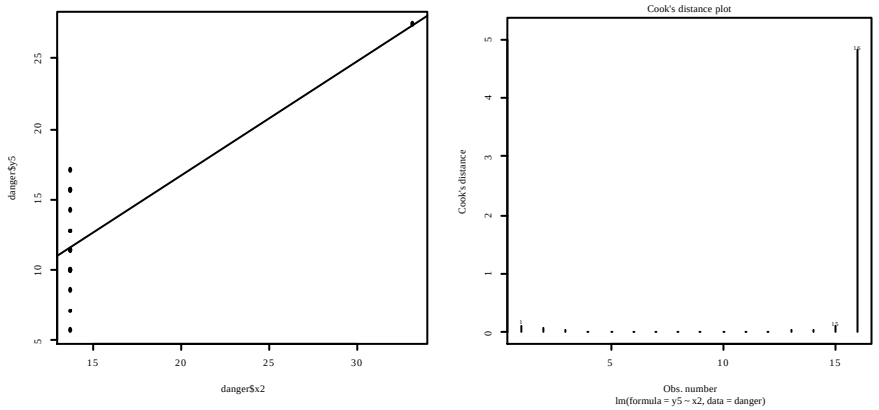


> lm4_lm(y4~x1,data=danger) *Variance non constante*




```
> coefficients(lm1)
(Intercept)          x1
     0.5221      0.8085
> coefficients(lm2)
(Intercept)          x1
     0.5237      0.8085
> coefficients(lm3)
(Intercept)          x1
     0.5201      0.8087
> coefficients(lm4)
(Intercept)          x1
     0.5200      0.8087
```

> lm5_lm(y5~x2,data=danger) *Point pivot*




```
> coefficients(lm5)
(Intercept)          x2
     0.5190      0.8087
```

2. Analyse de variance

2.1. Un facteur contrôlé

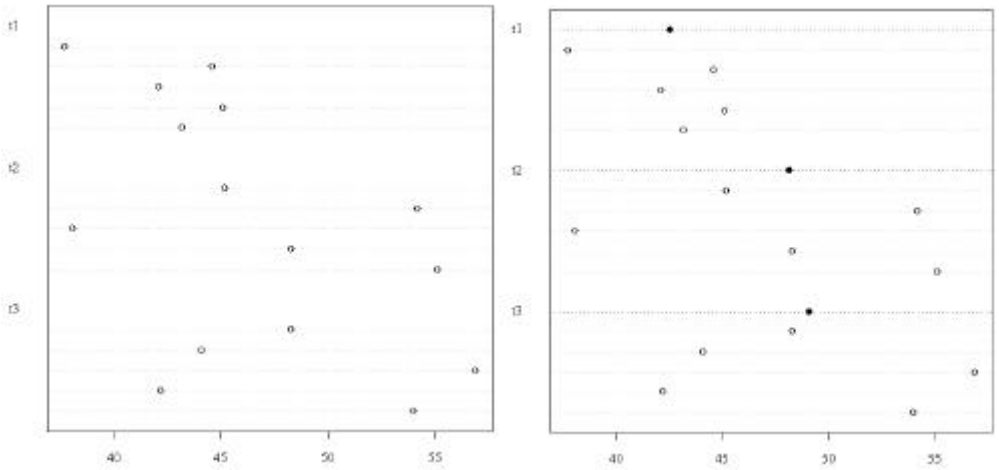
Les exemples sont ceux de Dagnélie, P. (1981) Théorie et méthodes statistiques. Exercices. Les Presses Agronomiques de Gembloux, Gembloux, 186 p.

Exercice 14.1 p. 102 - Quinze veaux ont été répartis au hasard en trois lots, alimentés chacun d'une façon différente. Les gains de poids observés au cours d'une même période et exprimés en kg étant les suivants, peut-on admettre qu'il n'y a pas de relation entre l'alimentation et la croissance des veaux ?

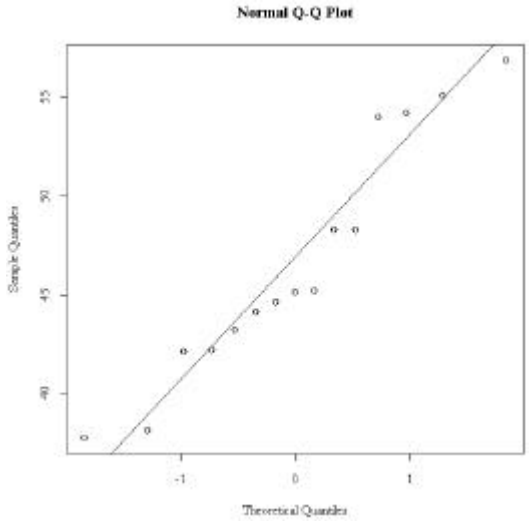
Alimentation		
1	2	3
37.7	45.2	48.3
44.6	54.2	44.1
42.1	38.1	56.9
45.1	48.3	42.2
43.2	55.1	54

Présenter les données sous la forme du lien entre un facteur et une réponse :

```
> ali_rep(c("t1","t2","t3"),c(5,5,5))
> ali
 [1] "t1" "t1" "t1" "t1" "t1" "t2" "t2" "t2" "t2" "t2" "t3" "t3" "t3" "t3" "t3"
> is.factor(ali)
[1] FALSE
> is.character(ali)
[1] TRUE
> ali_as.factor(ali)
> is.factor(ali)
[1] TRUE
> levels(ali)
[1] "t1" "t2" "t3"
> gain_scan()
1: 37.7
2: 44.6
3: 42.1
4: 45.1
…
14: 42.2
15: 54.0
16:
Read 15 items
dotplot(gain,gr=ali)
```



```
> tapply(gain,ali,mean)
   t1    t2    t3
42.54 48.18 49.10
> mgain_tapply(gain,ali,mean)
> dotplot(gain,gr=ali,gdata=mgain,gpch=16)
> anova(lm(gain~ali))
Analysis of Variance Table

Response: gain
          Df Sum Sq Mean Sq F value Pr(>F)
ali        2    126      63    1.95   0.18
Residuals 12    388      32

> qqnorm(gain)
> qqline(gain)
```



Le test est non significatif et l'ensemble des données peut être considéré comme un échantillon aléatoire simple d'une loi normale.

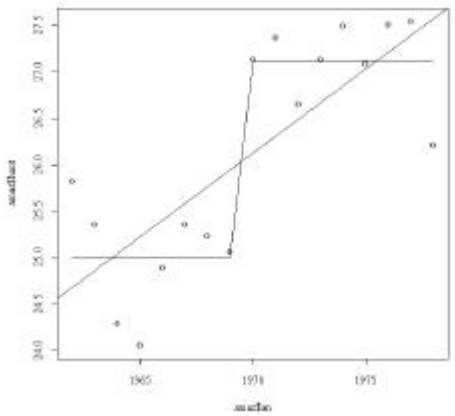
2.2. Unité entre analyse de variance et régression simple

an	haut	bas
1962	25.82	18.24
1963	25.35	16.5
1964	24.29	20.26
1965	24.05	20.97
1966	24.89	19.43
1967	25.35	19.31
1968	25.23	20.85
1969	25.06	19.54
1970	27.13	20.49
1971	27.36	21.91
1972	26.65	22.51
1973	27.13	18.81
1974	27.49	19.42
1975	27.08	19.1
1976	27.51	18.8
1977	27.54	18.8
1978	26.21	17.57

Un des aspects les plus frappants de l'hydrologie de la haute Amazone est la fluctuation saisonnière marquée du niveau des eaux. Les niveaux annuels des hautes et basses eaux ont été relevés de 1962 à 1978 à Iquitos. haut = Hautes eaux (m). bas = Basses Eaux (m).

A partir de 1970, l'ouverture de routes dans la haute vallée de l'Amazone a autorisé une déforestation à large échelle. Cette pratique est susceptible d'avoir des conséquences climatologiques et hydrologiques importantes. Ces conséquences sont-elles perceptibles dans les données ci-dessus ?

```
> amaz
     an  haut   bas
1  1962 25.82 18.24
2  1963 25.35 16.50
3  1964 24.29 20.26
4  1965 24.05 20.97
5  1966 24.89 19.43
6  1967 25.35 19.31
7  1968 25.23 20.85
8  1969 25.06 19.54
9  1970 27.13 20.49
10 1971 27.36 21.91
11 1972 26.65 22.51
12 1973 27.13 18.81
13 1974 27.49 19.42
14 1975 27.08 19.10
15 1976 27.51 18.80
16 1977 27.54 18.80
17 1978 26.21 17.57

> anova(lm(haut~an,data=amaz))
Analysis of Variance Table

Response: haut
          Df Sum Sq Mean Sq F value  Pr(>F)
an         1  13.35   13.35    20.9 0.00037 ***
Residuals 15   9.60    0.64
---
Signif. codes:  0  `***'  0.001  `**'  0.01  `*'  0.05  `.'  0.1  ` '  1

> plot(amaz$an,amaz$haut)
> abline(lm(haut~an,data=amaz))
> lines(amaz$an,predict(lm(haut~an>=1970,data=amaz)))
```



```
> anova(lm(haut~an>=1970,data=amaz))
Analysis of Variance Table

Response: haut
           Df Sum Sq Mean Sq F value  Pr(>F)
an >= 1970  1  18.99   18.99    71.8 4.2e-07 ***
Residuals  15   3.97    0.26
---
Signif. codes:  0  `***'  0.001  `**'  0.01  `*'  0.05  `.'  0.1  ` '  1
```

*Apporter des arguments pour choisir un modèle. Noter que l'assertion «*A partir de 1970, l'ouverture de routes …» *justifie* ***à priori*** *l'usage de l'hypothèse alternative* « avant et après sont différents ».

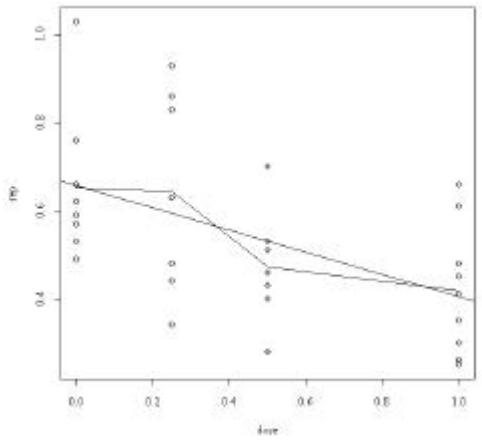
Chez le rat, on teste l'effet de l'ouabaï ne sur la teneur en noradrénaline du myocarde. Les résultats sont dans le tableau ci-dessous. On note x la dose d'ouabaï ne injectée et y la teneur en noradrénaline.

Ouabaïne (mg/kg)

0	**0.25**	**0.5**	**1**
0.49	0.63	0.51	0.66
0.66	0.93	0.53	0.48
0.59	0.48	0.28	0.25
0.62	0.34	0.7	0.3
0.76	0.83	0.43	0.35
0.57	0.44	0.4	0.61
0.62	0.86	0.46	0.45
0.53			0.26
1.03			0.41

```
> myo_read.table("myo.txt",h=T)
> myo[c(1:4,29:32),]
   dose  rep
1     0 0.49
2     0 0.66
3     0 0.59
4     0 0.62
29    1 0.61
30    1 0.45
31    1 0.26
32    1 0.41

> attach(myo)
> search()
[1] ".GlobalEnv"    "myo"           "package:mva"   "package:ctest"
[5] "Autoloads"     "package:base"

> plot(dose,rep)
```

```
> abline(lm(rep~dose))
lines(dose,predict(lm(rep~as.factor(dose)))
```



```
> lm1_lm(rep~dose)
> lm2_lm(rep~dose.fac)
> anova(lm1,lm2)
Analysis of Variance Table

Model 1: rep ~ dose
Model 2: rep ~ dose.fac
  Res.Df Res.Sum Sq Df Sum Sq F value Pr(>F)
1     30      0.849
2     28      0.805  2  0.043    0.75   0.48
> anova(lm2,lm1)
Analysis of Variance Table

Model 1: rep ~ dose.fac
Model 2: rep ~ dose
  Res.Df Res.Sum Sq Df Sum Sq F value Pr(>F)
1     28      0.805
2     30      0.849 -2 -0.043    0.75   0.48
> lm1_lm(rep~dose)
> lm2_lm(rep~dose.fac)
> lm1

Call:
lm(formula = rep ~ dose)

Coefficients:
(Intercept)         dose
      0.658       -0.253

> lm1_lm(rep~dose)
> lm2_lm(rep~dose.fac)
> summary(lm1)

Call:
lm(formula = rep ~ dose)

Residuals:
    Min      1Q  Median      3Q     Max
-0.2549 -0.1182 -0.0381  0.0813  0.3719

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.6581     0.0452    14.6  3.8e-15 ***
dose         -0.2525     0.0764    -3.3   0.0025 **
---
Signif. codes:  0  `***'  0.001  `**'  0.01  `*'  0.05  `.'  0.1  ` '  1
```

```
Residual standard error: 0.168 on 30 degrees of freedom
Multiple R-Squared: 0.267,      Adjusted R-squared: 0.242
F-statistic: 10.9 on 1 and 30 degrees of freedom,       p-value: 0.00248

> summary(lm2)

Call:
lm(formula = rep ~ dose.fac)

Residuals:
    Min      1Q  Median      3Q     Max
-0.3043 -0.1197 -0.0233  0.0728  0.3778

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.65222    0.05653   11.54  3.7e-12 ***
dose.fac0.25 -0.00794    0.08547   -0.09   0.9267
dose.fac0.5  -0.17937    0.08547   -2.10   0.0450 *
dose.fac1    -0.23333    0.07995   -2.92   0.0069 **
---
Signif. codes:  0  `***'  0.001  `**'  0.01  `*'  0.05  `.'  0.1  ` '  1

Residual standard error: 0.17 on 28 degrees of freedom
Multiple R-Squared: 0.304,      Adjusted R-squared: 0.23
F-statistic: 4.08 on 3 and 28 degrees of freedom,       p-value: 0.016

> anova(lm1,lm2)
Analysis of Variance Table
Model 1: rep ~ dose
Model 2: rep ~ dose.fac
  Res.Df Res.Sum Sq Df Sum Sq F value Pr(>F)
1     30      0.849
2     28      0.805  2  0.043    0.75   0.48

> detach("myo")
```

Autre exemple :

On a mesuré la quantité de sécrétions hormonales en fonction de la taille fixée chez des bars. Les résultats sont les suivants :

45	**55**	**65**	**75**
3.29	3.22	3.44	4.02
3.35	3.56	3.35	3.58
3.15	3.44	3.47	3.55
3.5	4.38	3.57	4.2
3.56	3.34	4.02	3.57
		3.51	4.3
			4.1

Analyser ces données.

On a mesuré dans 4 régions fixées, les poids en kg de cerfs mâles. Les régions A et C sont situées dans le nord de la France, la région B à l'est et la région D à l'ouest.

RegionA	**RegionB**	**RegionC**	**RegionD**
60.5	72.1	62	40.1
62.1	70.7	60.3	36.5
57.3	72.5	57.5	39.7
55	68	61.8	42.3
64.2	67.4	62.5	45.7
61.1	72.6	61.2	41.4
60	67.2	64.5	40.6
59.7	68.9	56.3	39.8
60.2	74.2	63.1	42
59.9	71.4	60.8	42.9

```
> cerf
    poi reg
1  60.5  rA
2  62.1  rA
…
39 42.0  rD
40 42.9  rD
```

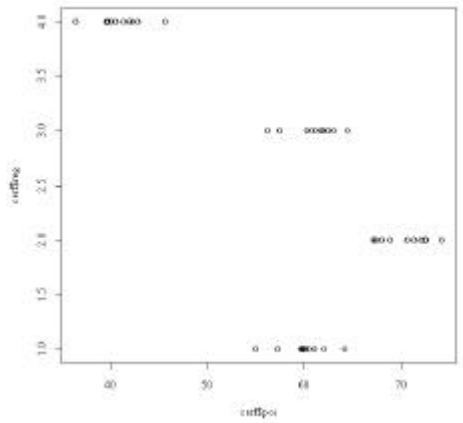
Pour l'homogénéité des variances :

```
> tapply(cerf$poi,cerf$reg,var)
   rA    rB    rC    rD
6.216 6.091 6.162 5.867

> library(ctest)
> bartlett.test(cerf$poi,cerf$reg)

         Bartlett test for homogeneity of variances

data:  cerf$poi and cerf$reg
Bartlett's K-square = 0.0083, df = 3, p-value = 0.9998 !!!
```

dans la documentation de bartlett.test :

See Also:

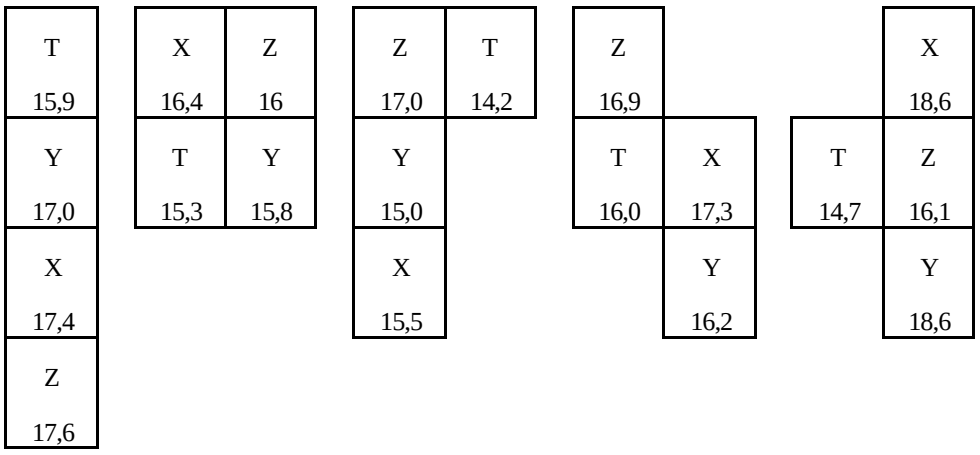
`var.test' for the special case of comparing variances in two samples from normal distributions; `fligner.test' for a rank-based (nonparametric) k-sample test for homogeneity of variances; `ansari.test' and `mood.test' for two rank based two-sample tests for difference in scale.



Doit-on faire un test ?

2.3. Deux facteurs

On veut tester l'efficacité de trois insecticides X, Y et Z contre la pyrale (papillon) du maï s. 5 champs non contigus de forme variable sont subdivisés en 4 parcelles de même surface et sur chacune on teste un insecticide ou rien (témoin T). Pour chaque parcelle, on mesure le poids de grains en kg d'une nombre constant de plants.

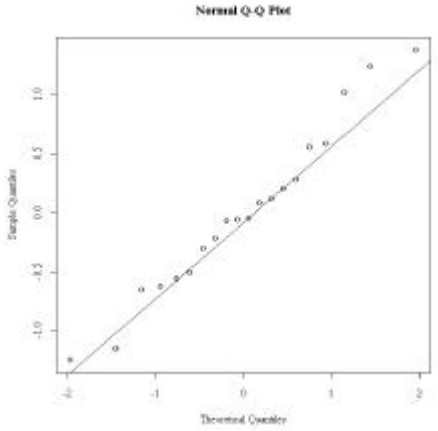


Champ 1 2 3 4 5

```
> insect
   champ prod  rep
1     c1    T 15.9
2     c1    X 17.0
3     c1    Y 17.4
4     c1    Z 17.6
5     c2    T 15.3
6     c2    X 16.4
7     c2    Y 15.8
8     c2    Z 16.0
9     c3    T 14.2
10    c3    X 15.5
11    c3    Y 15.0
12    c3    Z 17.0
13    c4    T 16.0
14    c4    X 17.3
15    c4    Y 16.2
16    c4    Z 16.9
17    c5    T 14.7
18    c5    X 18.6
19    c5    Y 18.6
20    c5    Z 16.1

> anova(lm(rep~prod+champ,data=insect))
Analysis of Variance Table

Response: rep
          Df Sum Sq Mean Sq F value Pr(>F)
prod       3   9.23    3.08    3.84  0.039 *
champ      4   7.81    1.95    2.44  0.104
Residuals 12   9.61    0.80
---
Signif. codes:  0  `***'  0.001  `**'  0.01  `*'  0.05  `.'  0.1  ` '  1
> anova(lm(rep~champ+prod,data=insect))
Analysis of Variance Table

Response: rep
          Df Sum Sq Mean Sq F value Pr(>F)
champ      4   7.82    1.95    2.44  0.104
prod       3   9.23    3.08    3.84  0.039 *
Residuals 12   9.61    0.80
---
Signif. codes:  0  `***'  0.001  `**'  0.01  `*'  0.05  `.'  0.1  ` '  1
```

Discuter l'intérêt des plans orthogonaux.

```
> lminsect_lm(rep~champ+prod,data=insect)
> qqnorm(residuals(lminsect))
> qqline(residuals(lminsect))
```



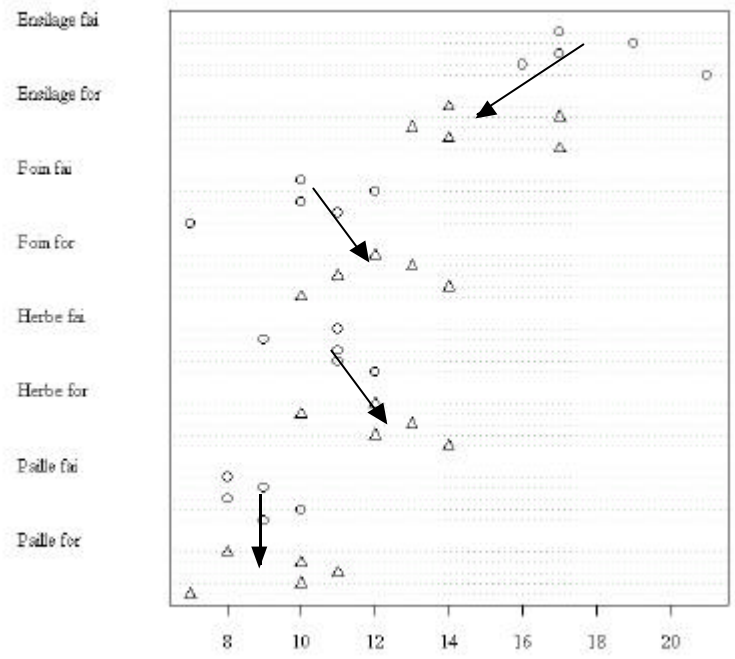

```
> summary(lminsect)

Call:
lm(formula = rep ~ champ + prod, data = insect)

Residuals:
    Min      1Q  Median      3Q     Max
-1.2450 -0.5225 -0.0525  0.3488  1.3750

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   15.820      0.566   27.95  2.7e-12 ***
champc2       -1.100      0.633   -1.74   0.1078
champc3       -1.550      0.633   -2.45   0.0306 *
champc4       -0.375      0.633   -0.59   0.5645
champc5        0.025      0.633    0.04   0.9691
prodX          1.740      0.566    3.07   0.0096 **
prodY          1.380      0.566    2.44   0.0313 *
prodZ          1.500      0.566    2.65   0.0212 *
---
Signif. codes:  0  `***'  0.001  `**'  0.01  `*'  0.05  `.'  0.1  ` '  1

Residual standard error: 0.895 on 12 degrees of freedom
Multiple R-Squared: 0.639,      Adjusted R-squared: 0.429
F-statistic: 3.04 on 7 and 12 degrees of freedom,       p-value: 0.0438

> contrasts(insect$champ)
   c2 c3 c4 c5
c1  0  0  0  0
c2  1  0  0  0
c3  0  1  0  0
c4  0  0  1  0
c5  0  0  0  1

> contrasts(insect$prod)
  X Y Z
T 0 0 0
X 1 0 0
Y 0 1 0
Z 0 0 1

> coefficients(lminsect)
(Intercept)     champc2     champc3     champc4     champc5       prodX
     15.820      -1.100      -1.550      -0.375       0.025       1.740
      prodY       prodZ
      1.380       1.500
> predict(lminsect)
    1     2     3     4     5     6     7     8     9    10    11    12
15.82 17.56 17.20 17.32 14.72 16.46 16.10 16.22 14.27 16.01 15.65 15.77
   13    14    15    16    17    18    19    20
15.44 17.18 16.82 16.94 15.84 17.58 17.22 17.34
```

Retrouver à la main une des valeurs prédites.

Dans une expérience sur le rendement des vaches laitières, on a choisi 40 animaux aussi identiques que possible et on les a répartis en 8 groupes de 5. Chaque groupe a été soumis à une alimentation différente.

	Aliment			
Dose	Paille	Foin	Herbe	Ensilage
forte	8	12	12	14
	10	13	10	17
	11	11	13	13
	10	14	12	14
	7	10	14	17
faible	8	10	11	17
	9	12	9	19
	8	10	11	17
	10	11	11	16
	9	7	12	21

```
> vache
   dose      ali rep
1   for   Paille   8
2   for   Paille  10
…
18  fai     Foin  10
19  fai     Foin  11
…
39  fai Ensilage  16
40  fai Ensilage  21

> anova(lm(rep~ali*dose,data=vache))
Analysis of Variance Table

Response: rep
          Df Sum Sq Mean Sq F value Pr(>F)
ali        3  305.0   101.7   40.07  6e-11 ***
dose       1    0.4     0.4    0.16 0.6940
ali:dose   3   37.4    12.5    4.91 0.0064 **
Residuals 32   81.2     2.5
---
Signif. codes:  0  `***'  0.001  `**'  0.01  `*'  0.05  `.'  0.1  ` '  1

> alidose_as.factor(paste(vache$ali,vache$dose))
> dotplot(vache$rep,gr=alidose)
```



Pour revenir sur la notion fondamentale d'interaction

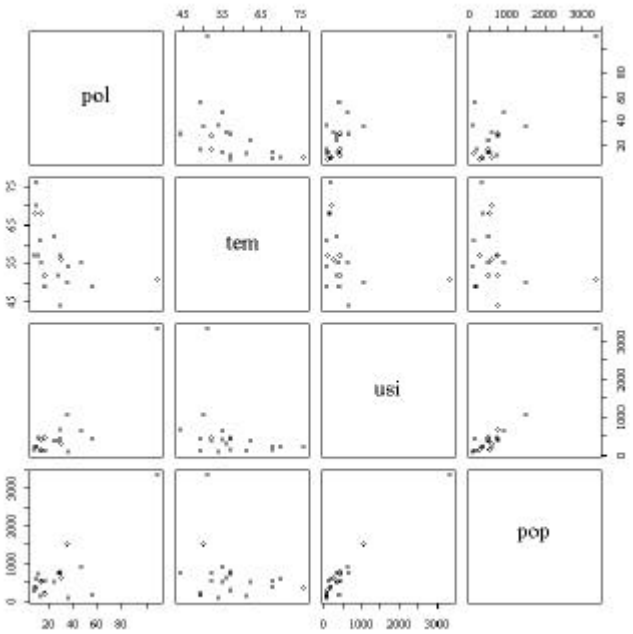
3. Régression multiple

Les données suivantes concernant 20 villes sont extraites d'une étude sur la pollution atmosphérique des villes des Etats-Unis :

	pol	**tem**	**usi**	**pop**
Atlanta	24	62	368	497
Baltimore	47	55	625	905
Chicago	110	51	3344	3369
Denver	17	52	454	515
Des_Moines	17	49	104	201
Detroit	35	50	1064	1513
Hartford	56	49	412	158
Indianapolis	28	52	361	746
Jacksonville	14	68	136	529
Kansas_City	14	55	381	507
Little_Rock	13	61	91	132
Louisville	30	56	291	593
Miami	10	76	207	335
Minneapolis	29	44	669	744
New_Orleans	9	68	204	361
Phoenix	10	70	213	582
San_Francisco	12	57	453	716
Washington	29	57	434	757
Wichita	8	57	125	277
Wilmington	36	54	80	80

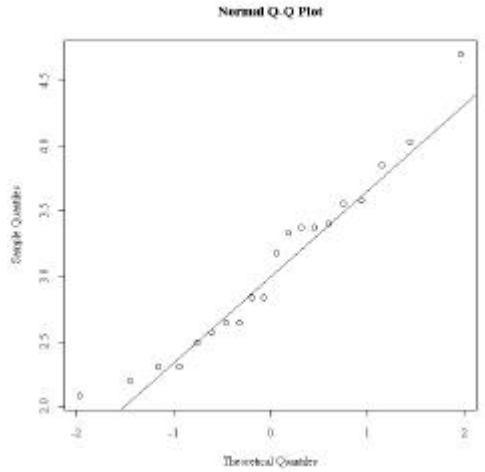
pol : teneur annuelle moyenne de l'air en SO2 en mg/m3
tem : Température annuelle moyenne en degrés Farenheit
usi : Nombre d'entreprises de plus de 20 personnes
pop : Population en milliers d'habitants (1970)

```
> pollu
           ville pol tem  usi  pop
1        Atlanta  24  62  368  497
2      Baltimore  47  55  625  905
3        Chicago 110  51 3344 3369
…
19       Wichita   8  57  125  277
20    Wilmington  36  54   80   80
> row.names(pollu)_pollu$ville
> pollu_pollu[,2:5]
> pollu
              pol tem  usi  pop
Atlanta        24  62  368  497
Baltimore      47  55  625  905
Chicago       110  51 3344 3369
Denver         17  52  454  515
Des_Moines     17  49  104  201
Detroit        35  50 1064 1513
Hartford       56  49  412  158
Indianapolis   28  52  361  746
Jacksonville   14  68  136  529
Kansas_City    14  55  381  507
Little_Rock    13  61   91  132
Louisville     30  56  291  593
Miami          10  76  207  335
Minneapolis    29  44  669  744
New_Orleans     9  68  204  361
Phoenix        10  70  213  582
San_Francisco  12  57  453  716
Washington     29  57  434  757
Wichita         8  57  125  277
Wilmington     36  54   80   80

> pairs(pollu)
```

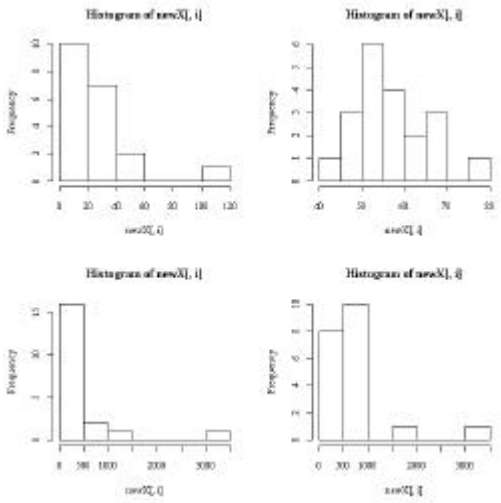


De très mauvaises distributions

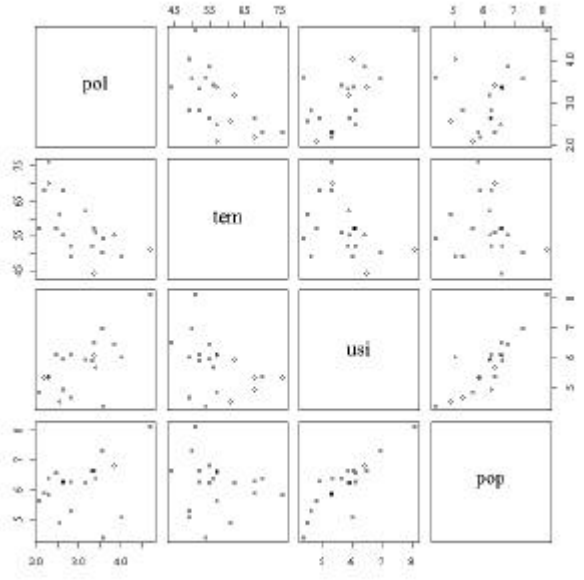
```
> qqnorm(log(pol))
qqline(log(pol))
```




```
> par(mfrow=c(2,2))
> apply(pollu,2,hist)
```




```
> pol_log(pollu$pol)
> usi_log(pollu$usi)
> pop_log(pollu$pop)
> tem_pollu$tem
> pairs(cbind.data.frame(pol,tem,usi,pop))
```



c'est beaucoup mieux

```
> lmpollu_lm(pol~tem+usi+pop)
> anova(lmpollu)
Analysis of Variance Table

Response: pol
          Df Sum Sq Mean Sq F value Pr(>F)
tem        1   3.43    3.43   15.87 0.0011 **
usi        1   1.72    1.72    7.94 0.0124 *
pop        1   0.43    0.43    1.99 0.1771
Residuals 16   3.46    0.22
---

> lmpollu_lm(pol~pop+usi+tem)
> anova(lmpollu)
Analysis of Variance Table

Response: pol
          Df Sum Sq Mean Sq F value  Pr(>F)
pop        1   1.26    1.26    5.83 0.02811 *
usi        1   3.76    3.76   17.38 0.00072 ***
tem        1   0.56    0.56    2.60 0.12632
Residuals 16   3.46    0.22
---

> lmpollu_lm(pol~usi+tem+pop)
> anova(lmpollu)
Analysis of Variance Table

Response: pol
          Df Sum Sq Mean Sq F value  Pr(>F)
usi        1   3.83    3.83   17.68 0.00067 ***
tem        1   1.33    1.33    6.13 0.02480 *
pop        1   0.43    0.43    1.99 0.17710
Residuals 16   3.46    0.22
---

> lmpollu_lm(pol~tem++pop+usi)
> anova(lmpollu)
Analysis of Variance Table

Response: pol
          Df Sum Sq Mean Sq F value Pr(>F)
tem        1   3.43    3.43   15.87 0.0011 **
```

```
pop        1   0.73    0.73    3.38 0.0845 .
usi        1   1.42    1.42    6.55 0.0210 *
Residuals 16   3.46    0.22
---
```

Voilà un problème sérieux.

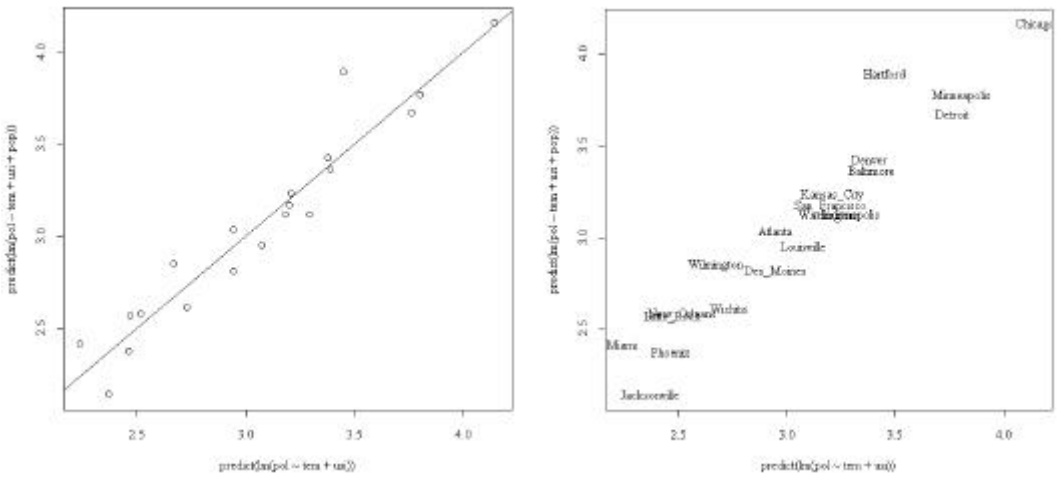
```
> cor(cbind.data.frame(tem,usi,pop))
        tem     usi     pop
tem  1.0000 -0.4102 -0.1495
usi -0.4102  1.0000  0.8598
pop -0.1495  0.8598  1.0000
```

On doit enlever une variable explicative redondante.

```
> cor(pol,pop)
[1] 0.3734
> cor(pol,tem)
[1] -0.6161
> cor(pol,usi)
[1] 0.6503

> coefficients (lm(pol~tem+usi+pop))
(Intercept)         tem         usi         pop
    2.78374    -0.02573     0.71712    -0.38543
```

Choisir d'enlever la source de contradiction. La pollution est corrélée positivement avec la population et cette variable intervient avec un coefficient négatif. Un signe opposé entre corrélation et coefficient de régression est l'indication d'une incohérence du modèle.

```
> plot(predict(lm(pol~tem+usi)),predict(lm(pol~tem+usi+pop)))
> abline(0,1)
> plot(predict(lm(pol~tem+usi)),predict(lm(pol~tem+usi+pop)),type="n")
> text(predict(lm(pol~tem+usi)),predict(lm(pol~tem+usi+pop)),row.names(pollu))
```



Enlever la variable pop ne modifie pas sensiblement le modèle.

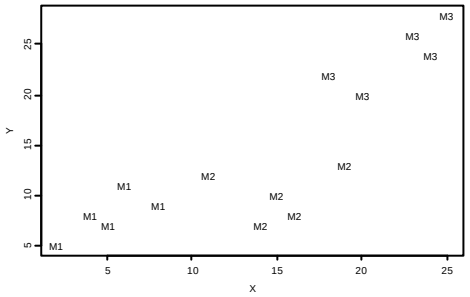
4. Analyse de covariance

L’exemple pédagogique est de J.D. Lebreton.

```
> M<-rep(c("M1","M2","M3"),c(5,5,5))
> M
 [1] "M1" "M1" "M1" "M1" "M1" "M2" "M2" "M2" "M2" "M2" "M3" "M3"
[13] "M3" "M3" "M3"
> X
 [1]  2  4  5  8  6 14 16 15 19 11 20 18 23 25 24
> Y
 [1]  5  8  7  9 11  7  8 10 13 12 20 22 26 28 24
```

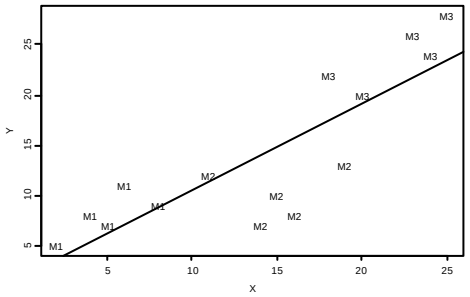
X niveau de départ, Y niveau d’arrivée, M méthode d’enseignement.

```
> covjdl<-cbind.data.frame(X,Y,M)
> names(covjdl)<-c("X","Y","M")
> covjdl
    X  Y  M
 1  2  5 M1
 2  4  8 M1
 3  5  7 M1
...
13 23 26 M3
14 25 28 M3
15 24 24 M3

> plot(X,Y,type="n")
> text(X,Y,M)
```



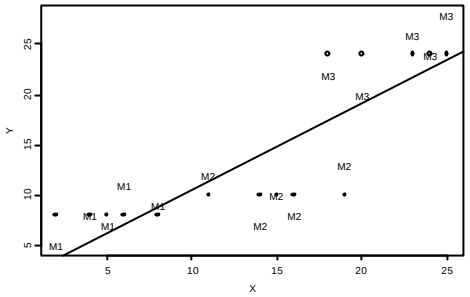
Une seule droite de régression

```
> lm1<-lm(Y~X)
> anova(lm1)
Analysis of Variance Table

Response: Y

Terms added sequentially (first to last)
          Df Sum of Sq Mean Sq F Value      Pr(F)
        X  1       599     599   31.53 0.00008407
Residuals 13       247      19
> abline(lm1)
```



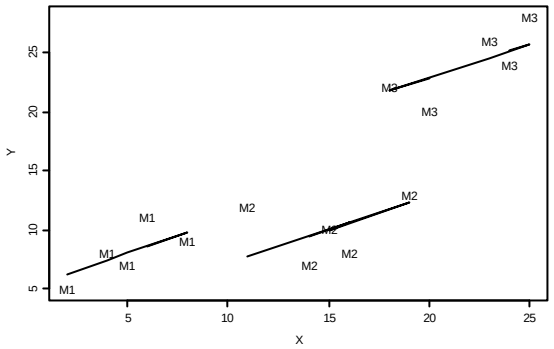
L'effet du facteur

```
> lm2<-lm(Y~M)
> anova(lm2)
Analysis of Variance Table

Response: Y

Terms added sequentially (first to last)
          Df Sum of Sq Mean Sq F Value      Pr(F)
        M  2       760   380.0   53.02 1.103e-006
Residuals 12        86     7.2
> points(X,predict(lm2))
```



Droites parallèles :

```
> lm3<-lm(Y~M+X)
> anova(lm3)
Analysis of Variance Table

Response: Y

Terms added sequentially (first to last)
          Df Sum of Sq Mean Sq F Value   Pr(F)
        M  2     760.0   380.0   72.58 0.00000
        X  1      28.4    28.4    5.43 0.03991
Residuals 11      57.6     5.2

> lm4<-lm(Y~X+M)
> anova(lm4)
Analysis of Variance Table

Response: Y

Terms added sequentially (first to last)
          Df Sum of Sq Mean Sq F Value     Pr(F)
        X  1     599.0   599.0   114.4 0.0000004
        M  2     189.4    94.7    18.1 0.0003329 Droites non égales
Residuals 11      57.6     5.2

> predict(lm3)
     1     2 3     4     5     6     7  8     9    10    11    12
 6.295 7.432 8 9.705 8.568 9.432 10.57 10 12.27 7.727 22.86 21.73
    13   14    15
 24.57 25.7 25.14
> predict(lm4)
     1     2 3     4     5     6     7  8     9    10    11    12
 6.295 7.432 8 9.705 8.568 9.432 10.57 10 12.27 7.727 22.86 21.73
    13   14    15
 24.57 25.7 25.14

> lines(X[M=="M1"],predict(lm3)[M=="M1"])
> lines(X[M=="M2"],predict(lm3)[M=="M2"])
> lines(X[M=="M3"],predict(lm3)[M=="M3"])
```




```
> coefficients(lm(Y[M=="M1"]~X[M=="M1"]))
 (Intercept) X[M == "M1"]
        4.25         0.75
> coefficients(lm(Y[M=="M2"]~X[M=="M2"]))
 (Intercept) X[M == "M2"]
       7.794       0.1471
> coefficients(lm(Y[M=="M3"]~X[M=="M3"]))
 (Intercept) X[M == "M3"]
       4.588       0.8824
```

5. Remise en question d'un modèle linéaire

Reprendre l'exemple introduit dans la fiche 2 (p. 2).

```
> ecrin[c(1:5,1314:1315),]
     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
1314  12  52   2   4
1315  12  52   1   7

> ric_ecrin$RIC
> sem_as.factor(ecrin$SEM)
> summary(sem)
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
18 25 26 28 28 25 28 28 27 26 26 28 28 23 26 26 27 27 26 24 24 27 24 27 27 24
27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52
24 28 24 23 22 25 23 21 26 27 25 28 21 27 24 26 24 24 26 25 25 23 25 28 25 23

> heu_as.factor(ecrin$HEU)
> levels(heu)_c("Mat","Soi")
> summary(heu)
Mat Soi
672 643

> sta_as.factor(ecrin$STA)
> summary(sta)
  1   2   3   4   5   6   7   8   9  10  11  12  13  14
   100  97  92  92  92  77 100  93  96  95  87  98  94 102
   101
> summary(ric)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00    4.00    7.00    7.48   10.00   24.00
```

lm, anova

La richesse dépend de l'heure :

```
> l1_lm(ric~heu)
> l1

Call:
lm(formula = ric ~ heu)

Coefficients:
(Intercept)       heuSoi
       8.97        -3.06

> summary(l1)

Call:
lm(formula = ric ~ heu)

Residuals:
   Min     1Q Median     3Q    Max
-8.972 -2.914 -0.914  2.086 15.028

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    8.972      0.148    60.6   <2e-16 ***
heuSoi        -3.057      0.212   -14.4   <2e-16 ***
---
Signif. codes:  0  `***'  0.001  `**'  0.01  `*'  0.05  `.'  0.1  ` '  1

Residual standard error: 3.84 on 1313 degrees of freedom
Multiple R-Squared: 0.137,      Adjusted R-squared: 0.136
F-statistic:  209 on 1 and 1313 degrees of freedom,     p-value:    0

> anova(l1)
Analysis of Variance Table

Response: ric
            Df Sum Sq Mean Sq F value Pr(>F)
heu          1   3071    3071     209 <2e-16 ***
Residuals 1313  19325      15
```

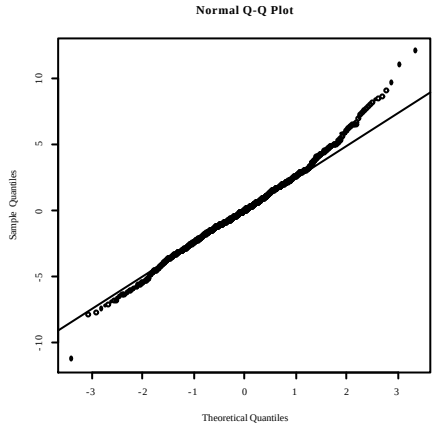
La richesse dépend de la semaine :

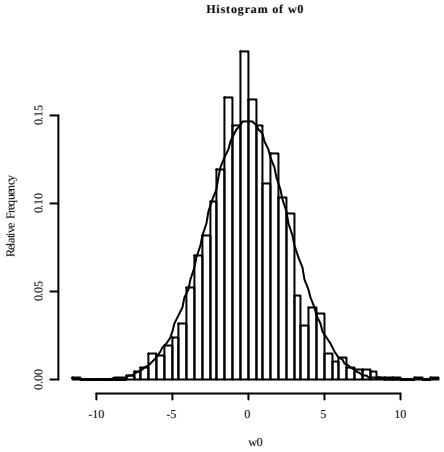
```
> l2_lm(ric~heu+sem)
> anova(l2)
Analysis of Variance Table

Response: ric
            Df Sum Sq Mean Sq F value Pr(>F)
heu          1   3071    3071   293.8 <2e-16 ***
sem         51   6133     120    11.5 <2e-16 ***
Residuals 1262  13192      10
```

La richesse dépend de la station :

```
> l3_lm(ric~heu+sem+sta)
> anova(l3)
Analysis of Variance Table

Response: ric
            Df Sum Sq Mean Sq F value Pr(>F)
heu.fac      1   3071    3071   396.6 <2e-16 ***
sem.fac     51   6133     120    15.5 <2e-16 ***
sta.fac     13   3519     271    35.0 <2e-16 ***
Residuals 1249   9673       8
```

```
> qqnorm (residuals(l3))
> qqline (residuals (l3))
```

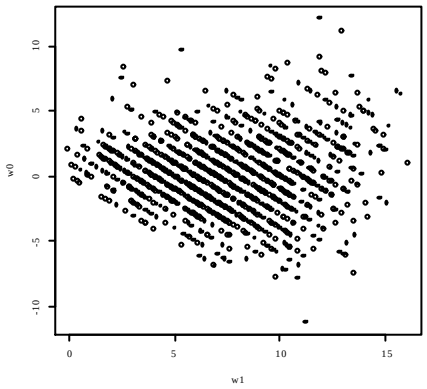





```
> w0_residuals(lm3)
> x0_seq(-10,10,le=100)
> hist(w0,proba=T,nclass=50)
> lines(x0,dnorm(x0,m=mean(w0),sd=sqrt(var(w0))))
```

Les résidus sont un peu sous-dispersés.

```
> w1_predict(lm3)
> plot(w1,w0)
```



La variable prédite est discrète. La variance ne semble pas constante.

```
> w2_cut(w1,c(-2,4,6,8,10,12,25))
```

Les valeurs prédites sont rangées en classes.

```
> table(w2)
w2
 (-2,4]   (4,6]   (6,8]  (8,10] (10,12] (12,25]
    188     240     321     280     177     109
```

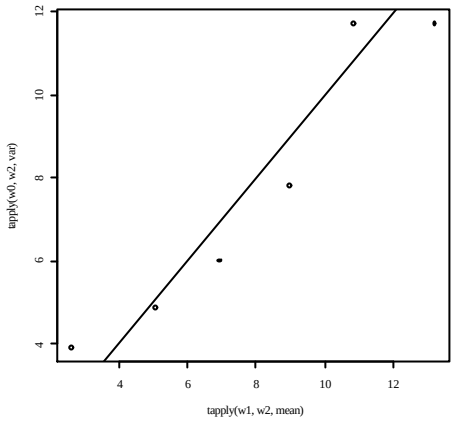
On calcule la moyenne des prédictions par classe de valeurs prédites :

```
> tapply(w1,w2,mean)
 (-2,4]   (4,6]   (6,8]  (8,10] (10,12] (12,25]
  2.655   5.075   6.953   9.007  10.846  13.224
```

On calcule la variance des résidus par classe de valeurs prédites :

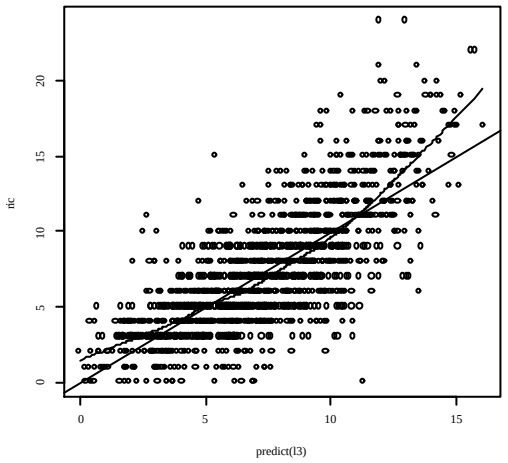
```
> tapply(w0,w2,var)
 (-2,4]   (4,6]   (6,8]  (8,10] (10,12] (12,25]
```

```
  3.887   4.881   5.998   7.793  11.736  11.731
> plot(tapply(w1,w2,mean),tapply(w0,w2,var))
> abline(0,1)
```



La variance des résidus croit sensiblement comme la moyenne des prédictions. Donc le modèle gaussien est invalide. Il faudrait dire : la richesse est une variable poissonienne (variance = moyenne) dont la moyenne est une fonction de l'heure, de la station et de la semaine. La liaison est aussi plus complexe que prévue :

```
> plot(predict(l3),ric)
> abline(0,1)
> lines(lowess(predict(l3),ric,f=0.3))
```



L'étude des résidus est toujours le point fondamental de l'interprétation. L'approche graphique des modèles statistiques : à consommer sans modération ...

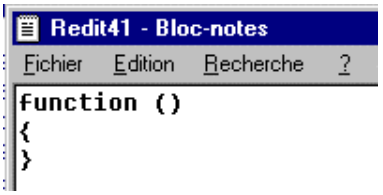
On y reviendra donc sur cet exemple dans le modèle linéaire généralisé.

6. Exercices

6.1. Approximation normale de la loi binomiale

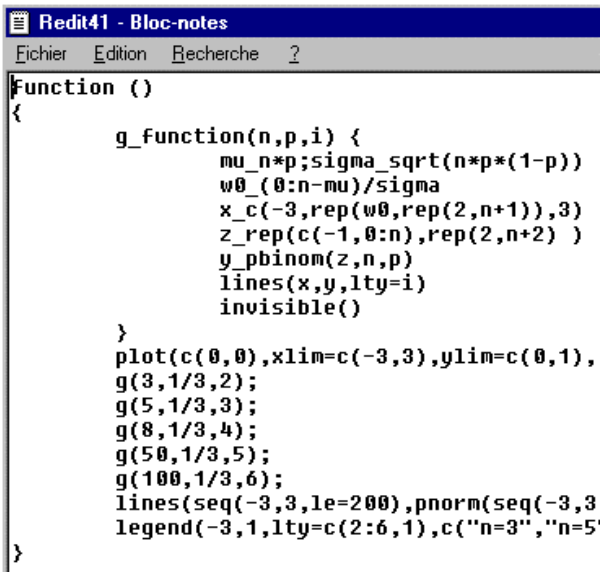
```
> fix(exo)
```

Une fenêtre est ouverte :



Remplir entre les deux accolades avec le texte :

```
g_function(n,p,i) {
      mu_n*p;sigma_sqrt(n*p*(1-p))
      w0_(0:n-mu)/sigma
      x_c(-3,rep(w0,rep(2,n+1)),3)
      z_rep(c(-1,0:n),rep(2,n+2) )
      y_pbinom(z,n,p)
      lines(x,y,lty=i)
      invisible()
}
plot(c(0,0),xlim=c(-3,3),ylim=c(0,1),  type="n",xlab="Convergence  vers  la  Loi
Normale",ylab="")
g(3,1/3,2);
g(5,1/3,3);
g(8,1/3,4);
g(50,1/3,5);
g(100,1/3,6);
lines(seq(-3,3,le=200),pnorm(seq(-3,3,le=200)),lty=1)
legend(-3,1,lty=c(2:6,1),c("n=3","n=5","n=8","n=50","n=100","loi normale"))
```



Fermer la fenêtre. On obtient l'objet exo qui est une fonction :

```
> exo
function ()
{
        g_function(n,p,i) {
                mu_n*p;sigma_sqrt(n*p*(1-p))
                w0_(0:n-mu)/sigma
...
        lines(seq(-3,3,le=200),pnorm(seq(-3,3,le=200)),lty=1)
        legend(-3,1,lty=c(2:6,1),c("n=3","n=5","n=8","n=50","n=100","loi
normale"))
}
```

Si on a fait une erreur, on a :

```
> fix(exo)
Error in edit(name, file, editor) : An error occurred on line 20
 use a command like
 x <- edit()
 to recover
```

Rappeler alors la dernière édition par :

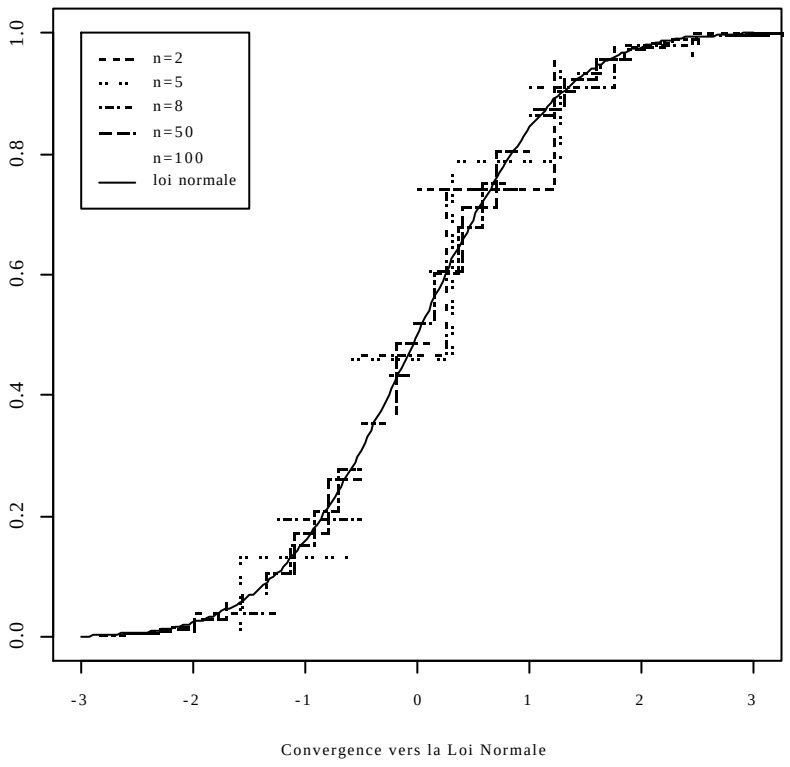
```
> exo_edit()
```

Si il reste encore une erreur, relancer :

```
Error in edit(name, file, editor) : An error occurred on line 21
 use a command like
 x <- edit()
 to recover
> exo_edit()
```

La fonction est alors exécutable :

```
> exo()
```



Si des erreurs subsistent en ayant une syntaxe correcte, revoir la fonction par :

```
fix(exo)
```

6.2. Edition de la loi binomiale

```
> a_matrix(0,15,14)
```

```
> for (i in 1:14) {a[1:(i+1),i]_round(1000*pbinom(0:i,i,1/2),digits=0)}
> a_as.data.frame(a);row.names(a)_0:14;names(a)_1:14
> a_matrix(0,29,14);
> for (j in 1:14) {i_j+14;a[1:(i+1),j]_round(1000*pbinom(0:i,i,1/2),digits=0)}
> a_as.data.frame(a);row.names(a)_0:28;names(a)_15:28
> a
```

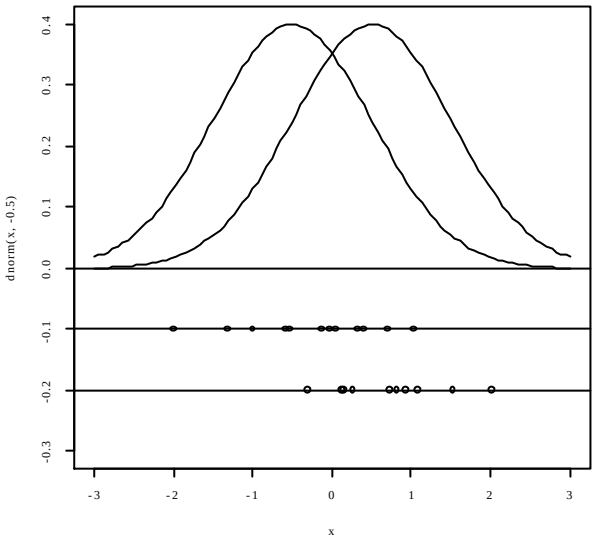
	15	16	17	18	19	20	21	22	23	24	25	26	27	28
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	4	2	1	1	0	0	0	0	0	0	0	0	0	0
3	18	11	6	4	2	1	1	0	0	0	0	0	0	0
4	59	38	25	15	10	6	4	2	1	1	0	0	0	0
5	151	105	72	48	32	21	13	8	5	3	2	1	1	0
6	304	227	166	119	84	58	39	26	17	11	7	5	3	2
7	500	402	315	240	180	132	95	67	47	32	22	14	10	6
8	696	598	500	407	324	252	192	143	105	76	54	38	26	18
9	849	773	685	593	500	412	332	262	202	154	115	84	61	44
10	941	895	834	760	676	588	500	416	339	271	212	163	124	92
11	982	962	928	881	820	748	668	584	500	419	345	279	221	172
12	996	989	975	952	916	868	808	738	661	581	500	423	351	286
13	1000	998	994	985	968	942	905	857	798	729	655	577	500	425
14	1000	1000	999	996	990	979	961	933	895	846	788	721	649	575
15	1000	1000	1000	999	998	994	987	974	953	924	885	837	779	714
16	0	1000	1000	1000	1000	999	996	992	983	968	946	916	876	828
17	0	0	1000	1000	1000	1000	999	998	995	989	978	962	939	908
18	0	0	0	1000	1000	1000	1000	1000	999	997	993	986	974	956
19	0	0	0	0	1000	1000	1000	1000	1000	999	998	995	990	982
20	0	0	0	0	0	1000	1000	1000	1000	1000	1000	999	997	994
21	0	0	0	0	0	0	1000	1000	1000	1000	1000	1000	999	998
22	0	0	0	0	0	0	0	1000	1000	1000	1000	1000	1000	1000
23	0	0	0	0	0	0	0	0	1000	1000	1000	1000	1000	1000
24	0	0	0	0	0	0	0	0	0	1000	1000	1000	1000	1000
25	0	0	0	0	0	0	0	0	0	0	1000	1000	1000	1000
26	0	0	0	0	0	0	0	0	0	0	0	1000	1000	1000
27	0	0	0	0	0	0	0	0	0	0	0	0	1000	1000
28	0	0	0	0	0	0	0	0	0	0	0	0	0	1000

A la ligne étiquetée *i* dans la colonne *n* on trouve $1000 \times P\left(X \leq i\right)$ pour une loi binomiale de paramètres *n* et 1/2.

6.3. Echantillons aléatoires simples

Pour illustrer ce que sont deux échantillons aléatoires simples :

```
> x_seq(-3,3,le =100)
> par (mfrow=c(1,1))
> plot(x,dnorm(x,-0.5),type="l",ylim=c(-0.3,0.4))
> lines(x,dnorm(x,0.5),type="l")
> y1_rnorm(12,-0.5)
> y2_rnorm(10,0.5)
> abline(h=c(0,-0.1,-0.2))
> points (y1,rep(-0.1,12))
> points(y2,rep(-0.2,10))
```



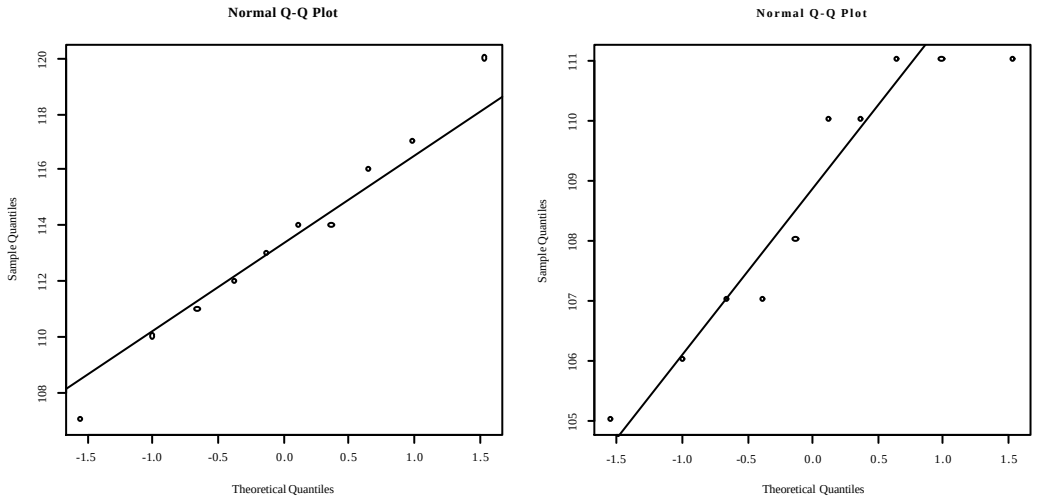
6.4. Comparer deux échantillons

Les exemples utilisés proviennent de l'ouvrage Hand, D.J., Daly, F., Lunn, A.D., McConway, K.J. & Ostrowski, E. (1994*) A handbook of small data sets.* Chapman & Hall, London. 1-458.

Situation 1 - La variable mesurée est la longueur de la mâchoire inférieure (en mm) de 10 chacals mâles et 10 chacals femelles (*Canis Aureus*) conservées au British Museum (Manly, B.F.J. (1991) *Randomization and Monte Carlo methods in biology*. Chapman & Hall, London. 1-281). La variable mesurée differe-t'elle entre les sexes dans cette espèce ? Taper les données :

```
> males_scan()
1: 120
2: 107
3: 110
4: 116
5: 114
6: 111
7: 113
8: 117
9: 114
10: 112
11:
Read 10 items
> femelles_scan()
1: 110
2: 111
3: 107
4: 108
5: 110
6: 105
7: 107
8: 106
9: 111
10: 111
11:
Read 10 items
> males
 [1] 120 107 110 116 114 111 113 117 114 112
> femelles
 [1] 110 111 107 108 110 105 107 106 111 111
```

Les variables sont-elles gaussiennes ?



```
> qqnorm(males)
> qqline(males)
> qqnorm(femelles)
> qqline(femelles)
```

Pour bien comprendre ce qui se passe :

```
> ?qqnorm
> print(qqnorm(males,plot.it=F))
$x
 [1]  1.5466353 -1.5466353 -1.0004905  0.6554235  0.1225808 -0.6554235
 [7] -0.1225808  1.0004905  0.3754618 -0.3754618

$y
 [1] 120 107 110 116 114 111 113 117 114 112
```

Les y sont les données. Les x sont des points-probabilités. L'idée est que 10 points équirépartis sur [0,1] forment neuf intervalles égaux. Mais le premier et le dernier ne sont pas sur les limites mais un peu en retrait. Ce retrait forment deux intervalles supplémentaires. On groupe ce qui est à gauche du premier et à droite du dernier pour former un dixième intervalle. Ceci est dans la fonction :

```
> ppoints(10,a=0.5)
 [1] 0.05 0.15 0.25 0.35 0.45 0.55 0.65 0.75 0.85 0.95
```

1/10 entre deux points consécutifs et le 1/10 restant également réparti à droite et à gauche. Les points probabilité sont définis par $p_i = \frac{i-0.5}{n}$. Pour les petits échantillons ($n \le 10$) on utilise la correction $p_i = \frac{i-3/8}{n+2/8}$.

```
> ppoints(10)
 [1] 0.06097561 0.15853659 0.25609756 0.35365854 0.45121951 0.54878049
 [7] 0.64634146 0.74390244 0.84146341 0.93902439
> (1-(3/8))/(10+(2/8))
[1] 0.06097561
```

L'essentiel est ailleurs :

```
> qnorm(ppoints(10))
 [1] -1.5466353 -1.0004905 -0.6554235 -0.3754618 -0.1225808  0.1225808
 [7]  0.3754618  0.6554235  1.0004905  1.5466353
```

Les x sont donc les quantiles théoriques $Q_t(p_i) = F^{-1}(p_i)$ où F est la fonction de répartition de la loi normale tandis que les y sont les quantiles observés aux mêmes points $Q_e(p_i)$. Si l'échantillon est tirée d'une loi normale on a, à l'erreur d'échantillonnage près, une droite. R n'est pas un logiciel de statistiques mais le logiciel d'une école de statistiques extraordinairement féconde. Trois références majeur : Chambers, J.M., Cleveland, W.S., Kleiner, B. & Tukey, P.A. (1983) *Graphical methods for data analysis*. Duxbury Press, Boston. 1-395. Cleveland, W.S. (1993) *Visualizing data.* Hobart Press, Summit, New Jersey. 1-360. Cleveland, W.S. (1994) *The elements of graphing data* AT&T Bell Laboratories, Murray Hill, New Jersey. 297 p.

Test de normalité

```
> library(ctest)
> ks.test(males,"pnorm",mean(males),sqrt(var(males)))
```

```
         One-sample Kolmogorov-Smirnov test

data:  males
D = 0.1359, p-value = 0.9927
alternative hypothesis: two.sided

> ks.test(femelles,"pnorm",mean(femelles),sqrt(var(femelles)))

         One-sample Kolmogorov-Smirnov test

data:  femelles
D = 0.2312, p-value = 0.6588
alternative hypothesis: two.sided
```

Rien ne s'oppose à la normalité des distributions.

```
> ks.test(males,femelles)
 [1] -0.1 -0.2 -0.3 -0.4 -0.5 -0.7 -0.6 -0.5 -0.3 -0.2 -0.1  0.0

         Two-sample Kolmogorov-Smirnov test

data:  males and femelles
D = 0.7, p-value = 0.01489
alternative hypothesis: two.sided

Warning message:
cannot compute correct p-values with ties in: ks.test(males, femelles)
```

Les deux distributions ne sont pas identiques.

```
> bartlett.test(list(males,femelles))

         Bartlett test for homogeneity of variances

data:  list(males, femelles)
Bartlett's K-square = 1.9942, df = 1, p-value = 0.1579
```

L'inégalité des variances n'est pas en cause.

```
> t.test(males,femelles)

         Welch Two Sample t-test

data:  males and femelles
t = 3.4843, df = 14.894, p-value = 0.00336
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 1.861895 7.738105
sample estimates:
mean of x mean of y
    113.4     108.6
```

L'égalité des moyennes est rejetée.

```
> wilcox.test(males,femelles)

         Wilcoxon rank sum test with continuity correction

data:  males and femelles
W = 87.5, p-value = 0.004845
alternative hypothesis: true mu is not equal to 0

Warning message:
Cannot compute exact p-value with ties in: wilcox.test(males, femelles)
```

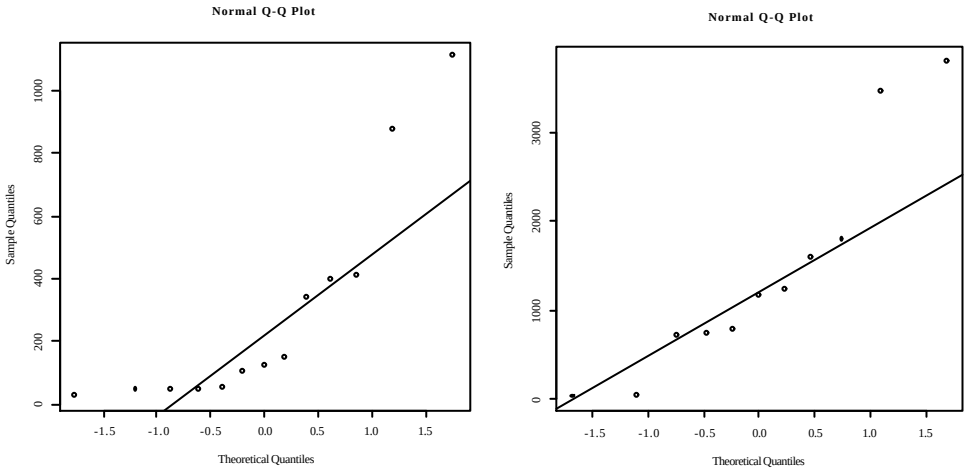
Le test non paramétrique confirme.

Et si vous voulez exactement savoir ce que vous faîtes, voici le source :

```
> bartlett.test
function (x, g)
{
    LM <- FALSE
    if (is.list(x)) {
        if (length(x) < 2)
            stop("x must be a list with at least 2 elements")
...

> t.test
function (x, y = NULL, alternative = c("two.sided", "less", "greater"),
    mu = 0, paired = FALSE, var.equal = FALSE, conf.level = 0.95)
{
    alternative <- match.arg(alternative)
    if (!missing(mu) && (length(mu) != 1 || is.na(mu)))
        stop("mu must be a single number")
...
```

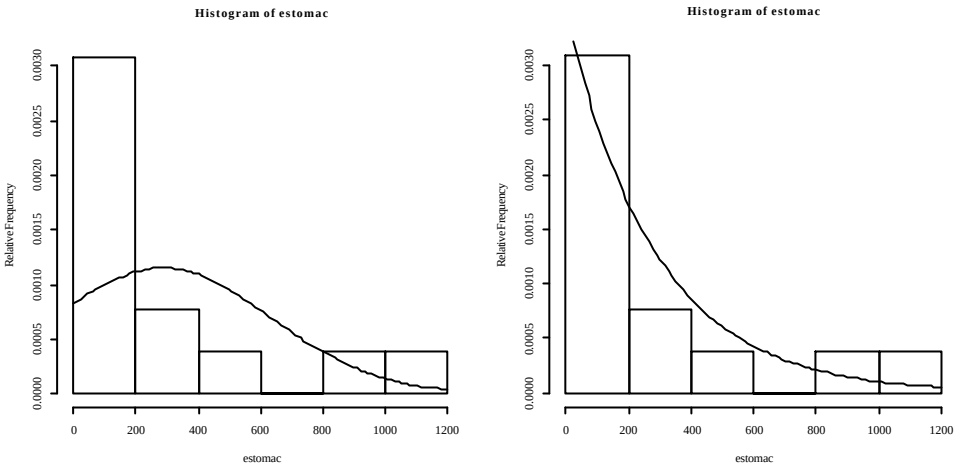
Situation 2 - La variable mesurée est le temps de survie (en jours) de patients atteints d'un cancer et traités avec un médicament donné (Cameron, E. & Pauling, L. (1978) Supplemental ascorbate in the supportive treatment of cancer: re-evaluation of prolongation of survival times in terminal human cancer. *Proceeding of the National Academy of Sciences of the USA* : 75, 4538-4542). Cette variable dépend elle du type de cancer ?

```
> estomac
 [1]  124   42   25   45  412   51 1112   46  103  876  146  340  396
> poumon
 [1] 1235   24 1581 1166   40  727 3808  791 1804 3460  719
> qqnorm(estomac)
> qqline(estomac)
> qqnorm(poumon)
> qqline(poumon)
```



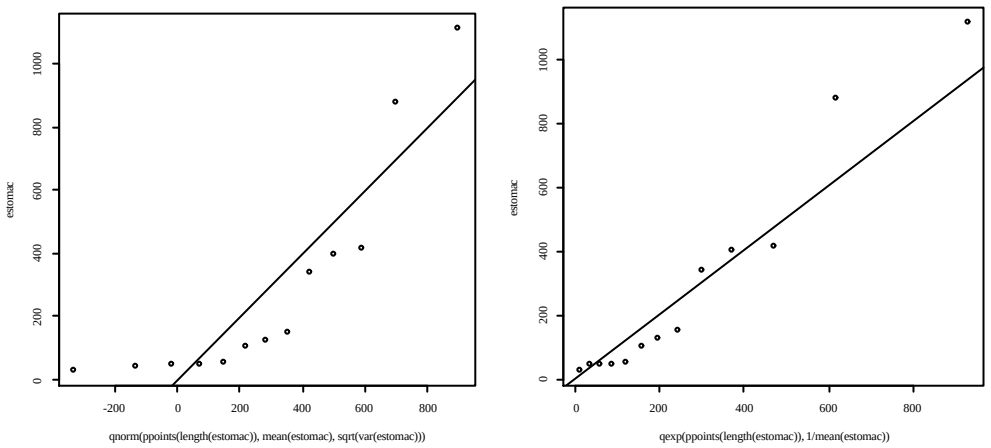
Pas normal du tout !

```
> hist(estomac,proba=T)
> lines(x0,dnorm(x0,mean(estomac),sqrt(var(estomac))))

> x0_seq(from=0,to=1200,le=100)

> hist(estomac,proba=T)
> lines(x0,dexp(x0,1/mean(estomac)))
```



R est un logiciel qui inspire franchement de la considération pour ses auteurs.

```
>
qqplot(qnorm(ppoints(length(estomac)),mean(estomac),sqrt(var(estomac))),estomac)
> abline(0,1)
> qqplot(qexp(ppoints(length(estomac)),1/mean(estomac)),estomac)
> abline(0,1)
```



Mieux mais pas tout à fait ça. De toute manière, les tests non paramétriques s'imposent :

```
> ks.test(estomac,poumon)

        Two-sample Kolmogorov-Smirnov test

data:  estomac and poumon
D = 0.6643, p-value = 0.01040
alternative hypothesis: two.sided

> wilcox.test(estomac,poumon)

        Wilcoxon rank sum test

data:  estomac and poumon
W = 31, p-value = 0.01836
alternative hypothesis: true mu is not equal to 0
```