TD5 : régression linéaire

BIO5XX - BIOSTATISTIQUE L3

23 octobre 2016

Objectifs de la séance
Régression linéaire sous R

A partir des données de peupliers utilisées en séance 2, on souhaite déterminer s'il est possible de prédire le poids d'un arbre à partir de sa hauteur et de son diamètre. En effet, ces arbres vigoureux à croissance rapide risquent un jour de constituer une source de remplacement en combustible et produits chimiques. Des études préliminaires ont montré l'existence d'une relation étroite entre le poids du bois sec des jeunes peupliers et une variable fonction du diamètre et de la taille des arbres (DDH=carré des diamètres multiplié par la hauteur).

On demande de déterminer si l'on peut utiliser le diamètre et la hauteur pour prédire de manière fiable le rendement en bois.

L'analyse sera réalisée sur les arbres âgés de 3 ans, plantés l'année 1, et n'ayant subi aucun traitement (arbres témoins : traitement 1).

Exemple R - 1 :

Lecture des données à partir du fichier *peuplier.txt*

```
peuplier<-read.table("peuplier.txt",header=TRUE)
names(peuplier)
```

```
## [1] "Site"       "Annee"      "Traitement" "Diametre"   "Hauteur"
## [6] "Poids"      "Age"
```

Exemple R - 2 :

Préparation du fichier de données : arbre âges de 3 ans, plantés l'année 1, ayant subis le traitement 1.

```
peuplier1<-peuplier[peuplier$Age==3 &
                    peuplier$Annee==1 &
                    peuplier$Traitement==1,]
peuplier1
```

```
##    Site Annee Traitement Diametre Hauteur Poids Age
## 1     1     1          1     2.23    3.76  0.17   3
## 2     1     1          1     2.12    3.15  0.15   3
## 3     1     1          1     1.06    1.85  0.02   3
## 4     1     1          1     2.12    3.64  0.16   3
## 5     1     1          1     2.99    4.64  0.37   3
```

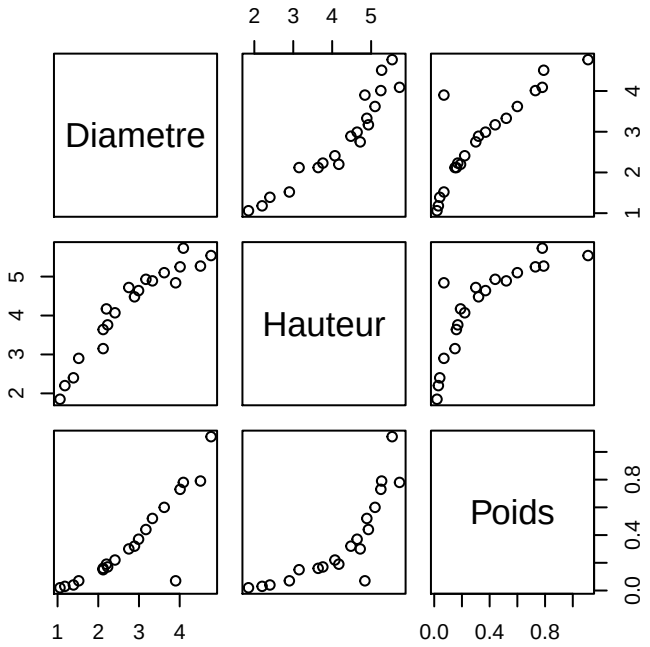
```
## 6     1     1          1     4.01    5.25  0.73   3
## 7     1     1          1     2.41    4.07  0.22   3
## 8     1     1          1     2.75    4.72  0.30   3
## 9     1     1          1     2.20    4.17  0.19   3
## 10    2     1          1     4.09    5.73  0.78   3
## 11    2     1          1     3.62    5.10  0.60   3
## 12    2     1          1     4.77    5.54  1.11   3
## 13    2     1          1     1.39    2.40  0.04   3
## 14    2     1          1     2.89    4.48  0.32   3
## 15    2     1          1     3.90    4.84  0.07   3
## 16    2     1          1     1.52    2.90  0.07   3
## 17    2     1          1     4.51    5.27  0.79   3
## 18    2     1          1     1.18    2.20  0.03   3
## 19    2     1          1     3.17    4.93  0.44   3
## 20    2     1          1     3.33    4.89  0.52   3
```

```
peuplier2<-peuplier1[,4:6]
peuplier2
```

```
##    Diametre Hauteur Poids
## 1      2.23    3.76  0.17
## 2      2.12    3.15  0.15
## 3      1.06    1.85  0.02
## 4      2.12    3.64  0.16
## 5      2.99    4.64  0.37
## 6      4.01    5.25  0.73
## 7      2.41    4.07  0.22
## 8      2.75    4.72  0.30
## 9      2.20    4.17  0.19
## 10     4.09    5.73  0.78
## 11     3.62    5.10  0.60
## 12     4.77    5.54  1.11
## 13     1.39    2.40  0.04
## 14     2.89    4.48  0.32
## 15     3.90    4.84  0.07
## 16     1.52    2.90  0.07
## 17     4.51    5.27  0.79
## 18     1.18    2.20  0.03
## 19     3.17    4.93  0.44
## 20     3.33    4.89  0.52
```

1 Mise en place de l'analyse

```
plot(peuplier2)
```



La variable poids semble bien corrélée avec les variables diamètre et hauteur. En réalité on va étudier la relation qui lie le poids à la variable ddh ($diamètre^2 \times hauteur$).

— Justifier cette relation.
— Quelle est la variable réponse (à expliquer) ?
— Quelle est la variable explicative ?

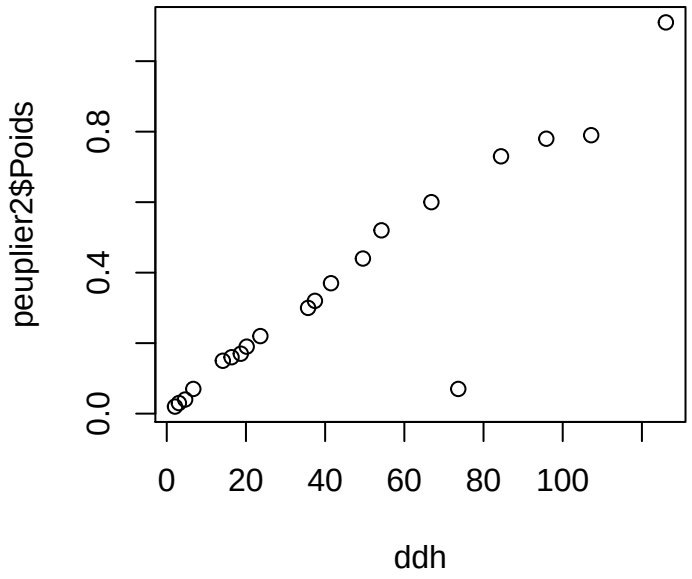
Exemple R - 3 :

Création de la nouvelle variable ddh

```
ddh<-peuplier2$Diametre*peuplier2$Diametre*peuplier2$Hauteur
ddh
```

```
##  [1]  18.69810  14.15736   2.07866  16.35962  41.48206  84.42052  23.63897
##  [8]  35.69500  20.18280  95.85201  66.83244 126.05107   4.63704  37.41741
## [15]  73.61640   6.70016 107.19233   3.06328  49.54108  54.22472
```

```
plot(ddh,peuplier2$Poids)
```



La relation semble linéaire. Un point semble bien à part. Il s'agit en fait d'une erreur dans les données. C'est l'observation 15 qui pour une valeur de ddh élevée (73.61) a un poids trop faible (0.07). Vous corrigerez cette valeur de Poids en mettent 0.7 à la place.

Exemple R - 4 :

Correction des données : l'observation 15 a un poids trop faible (0.07). Vous corrigerez cette valeur de Poids en mettent 0.7 à la place.

```
ddh
```

```
##  [1]  18.69810  14.15736   2.07866  16.35962  41.48206  84.42052  23.63897
##  [8]  35.69500  20.18280  95.85201  66.83244 126.05107   4.63704  37.41741
## [15]  73.61640   6.70016 107.19233   3.06328  49.54108  54.22472
```

```
peuplier2$Poids
```

```
##  [1] 0.17 0.15 0.02 0.16 0.37 0.73 0.22 0.30 0.19 0.78 0.60 1.11 0.04 0.32
## [15] 0.07 0.07 0.79 0.03 0.44 0.52
```

```
peuplier2[15,3] <- 0.7
```

Vérifier que le point a été bien enlevé sur le graphique.

On va donc chercher le modèle $Poids = a \times ddh + b + e$, ou e est l'erreur (les erreurs sont indépendantes, de variance constante et suivent une loi normale de moyenne nulle et de variance constante)

2 Calcul des paramètres du modèle

Exemple R - 5 :

Calcul des paramètres du modèle.

```
modele1<-lm(peuplier2$Poids~ddh)
modele1

##
## Call:
## lm(formula = peuplier2$Poids ~ ddh)
##
## Coefficients:
## (Intercept)          ddh
##     0.01999      0.00829
```

L'équation de la droite est donc : $Poids = 0.0082897 \times ddh + 0.0199907$.

3 Validation du modèle

3.1 Test sur la pente

H0 : $a = 0$
H1 : $a \neq 0$

Exemple R - 6 :

La p-value du test t sur la pente est quasi-nulle ($p < 2.10^{-16}$) : on accepte H1. Il y a bien une relation entre le Poids et ddh. ddh explique 98% de la variation du Poids. C'est relativement élevé.

```
summary(modele1)

##
## Call:
## lm(formula = peuplier2$Poids ~ ddh)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.118582 -0.015511  0.003375  0.010804  0.069752
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.019991   0.013650   1.464     0.16
## ddh         0.008290   0.000239  34.683   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0388 on 18 degrees of freedom
## Multiple R-squared:  0.9853,Adjusted R-squared:  0.9844
## F-statistic:  1203 on 1 and 18 DF,  p-value: < 2.2e-16
```

3.2 Etude des résidus

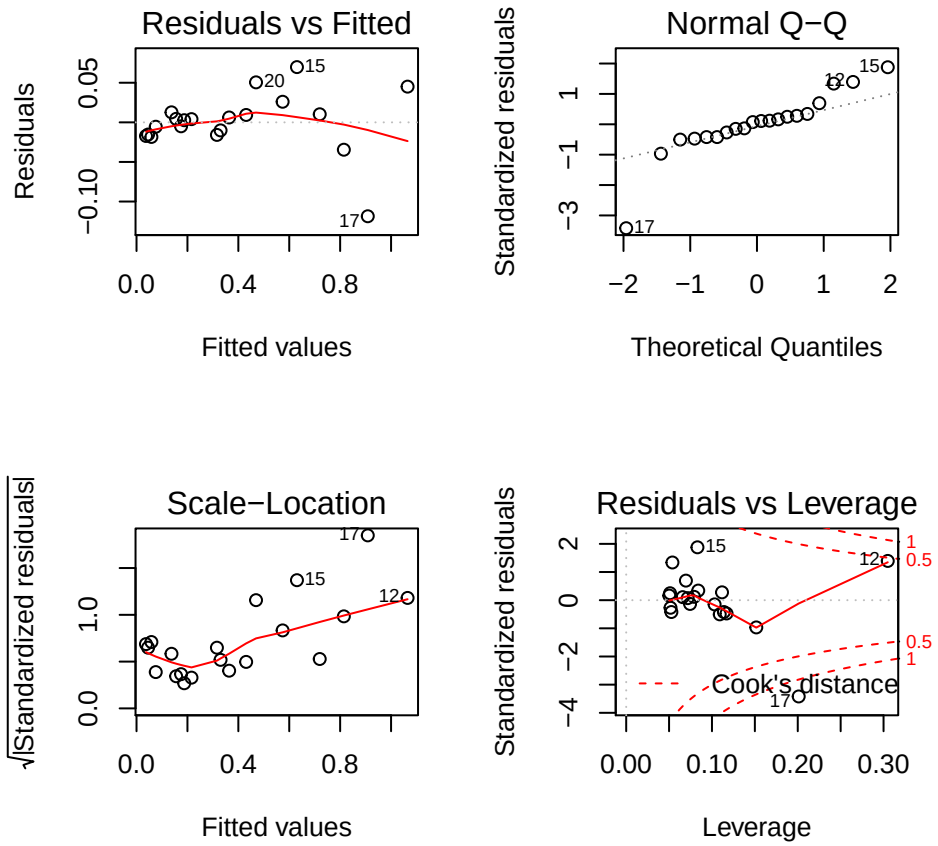
Exemple R - 7 :

Regarder les objets contenus dans lm. On peut par exemple appeler les résidus brutes

```
?lm
modele1$residuals
```

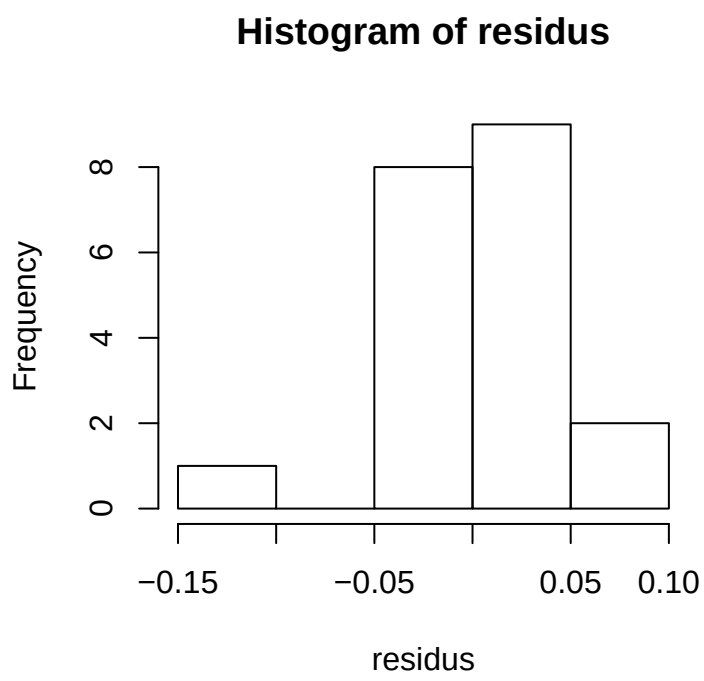
```
##            1            2            3            4            5
## -0.004992127  0.012649219 -0.017222140  0.004393207  0.006135978
##            6            7            8            9           10
##  0.010189581  0.004049666 -0.015891073  0.002700208 -0.034573876
##           11           12           13           14           15
##  0.025989298  0.045085430 -0.018430309 -0.010169296  0.069752394
##           16           17           18           19           20
## -0.005532928 -0.118581528 -0.015384331  0.009329284  0.050503342
```

```
par(mfrow=c(2,2))
plot(modele1)
```



Il s'agissait de vous montrer ce qu'il est possible de faire avec la fonction lm. Mais on va reprendre les graphiques dont on a besoin « à la main ». Regarder l'aide pour comprendre quels graphiques ont été tracés ici.

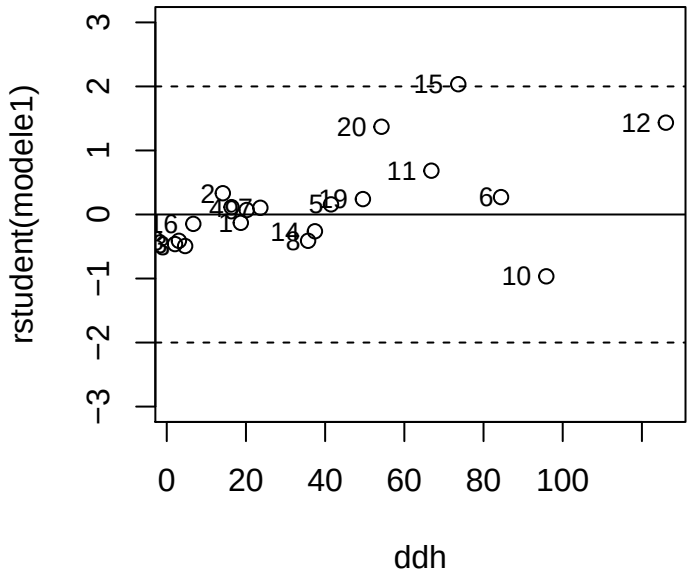
```
residus<-modele1$residuals
hist(residus)
```




```
shapiro.test(modele1$residuals)

##
##  Shapiro-Wilk normality test
##
## data:  modele1$residuals
## W = 0.86638, p-value = 0.01016
```

On ne peut pas accepter l'hypothèse de normalité au seuil de 5%.
On va neanmoins maintenant s'intéresser à la distribution des résidus studentisés en fonction de ddh.

```
plot(ddh,y=rstudent(modele1),ylim=c(-3,3))
text(x=ddh,y=rstudent(modele1),label=(1:20),adj=1.5,cex=0.8)
abline(+2,0,lty=2)
abline(0,0,lty=1)
abline(-2,0,lty=2)
```



4 Régression linéaire multiple

4.1 Étude de la processionnaire du pin

4.1.1 Description des données «processionnaire du pin»

33 observations ont été réalisées pour étudier les facteurs qui influencent la répartition des nids de processionnaire dans les pins. Il s'agit en fait de rechercher une relation entre les attaques de la processionnaire du pin et les diverses caractéristiques du peuplement forestier.

— Unité de l'étude : parcelle de 10 ha
— Sous unités : placettes de 5 ha.

1. Altitude (m)
2. Pente (en °)
3. Densité (Nombre de pins/placette de 5 ha)
4. Hauteur (en m)
5. Diamètre du tronc (en cm)
6. Nombre moyen de nids de processionnaire par arbre

Objectif

Recherche d'une relation linéaire entre les attaques de processionnaire du pin et diverses caractéristiques du peuplement forestier (altitude, pente, densité, hauteur, diamètre du tronc)...

4.2 Etude préliminaire

Calculer les moyennes et les variances des différentes variables.

Exemple R - 8 :

Lecture des données à partir du fichier *pin.txt*

```
pin<-read.table("pin.txt",h=T)
names(pin)

## [1] "alt"   "pente" "densi" "haut"  "diam"  "proce"

colMeans(pin)

##          alt        pente        densi         haut         diam
## 1315.3333333   28.7272727   11.4545455    4.4515152   15.2515152
##        proce
##    0.8112121

var(pin)

##               alt      pente      densi       haut        diam       proce
## alt   16650.54167 133.687500 661.500000 43.1166667 157.5479167 -55.1604167
## pente   133.68750  58.267045  25.377841  1.1176136   3.2051136  -2.9752841
## densi   661.50000  25.377841  90.943182  4.1133523  12.1008523  -4.3355682
## haut     43.11667   1.117614   4.113352  1.0832008   4.0510133  -0.3003144
## diam    157.54792   3.205114  12.100852  4.0510133  18.5119508  -0.5472831
## proce   -55.16042  -2.975284  -4.335568 -0.3003144  -0.5472831   0.6500047
```

Calculer la corrélation entre toutes les variables

Exemple R - 9 :

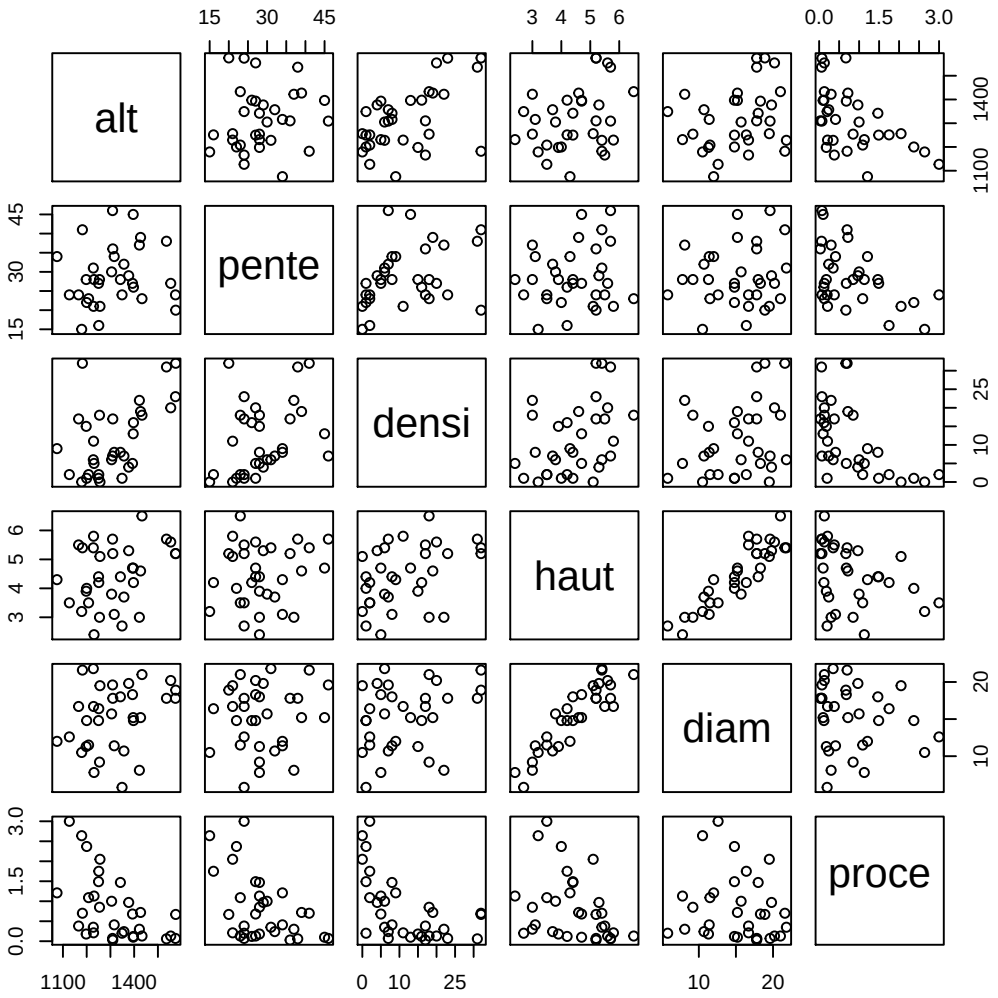
Corrélations pour chaque paire de variables.

```
cor(pin)

##              alt       pente      densi       haut        diam      proce
## alt    1.0000000  0.13572665  0.5375645  0.3210528  0.28377386 -0.5302188
## pente  0.1357266  1.00000000  0.3486247  0.1406778  0.09759012 -0.4834579
## densi  0.5375645  0.34862466  1.0000000  0.4144349  0.29492039 -0.5639008
## haut   0.3210528  0.14067785  0.4144349  1.0000000  0.90465522 -0.3579014
## diam   0.2837739  0.09759012  0.2949204  0.9046552  1.00000000 -0.1577712
## proce -0.5302188 -0.48345792 -0.5639008 -0.3579014 -0.15777120  1.0000000

cor.test(pin$haut,pin$diam)

##
##  Pearson's product-moment correlation
##
## data:  pin$haut and pin$diam
## t = 11.82, df = 31, p-value = 5.125e-13
## alternative hypothesis: true correlation is not equal to 0
```

```
## 95 percent confidence interval:
##  0.8142253 0.9522256
## sample estimates:
##       cor
## 0.9046552
```

A priori il faudrait ne prendre que hauteur ou que diamètre. Dans un premier temps on va prendre toutes les variables.

```
plot(pin)
```



L'objectif de l'étude est donc d'étudier la relation entre la variable réponse y (nombre d'attaques par la processionnaire du pin représentée par le nombre moyen de nids de processionnaire par arbre), en fonction des autres variables.

4.3 Mise en place de la régression multiple

— Énoncer les différentes hypothèses du modèle de régression multiple
— Étudier la relation entre y (nombre d'attaques) et toutes les variables explicatives.

Exemple R - 10 :

Mise en place du modèle.

```
lm1<-with(pin,
          lm(proce~alt+pente+densi+haut+diam)
         )
```

Exemple R - 11 :

Mise en place du modèle.

```
summary(lm1)
```

```
##
## Call:
## lm(formula = proce ~ alt + pente + densi + haut + diam)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.97612 -0.35102 -0.08102  0.30525  1.24803
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.6785969  1.2218580   4.648 7.85e-05 ***
## alt         -0.0022843  0.0008999  -2.538  0.01722 *
## pente       -0.0366575  0.0135155  -2.712  0.01149 *
## densi       -0.0105973  0.0135818  -0.780  0.44203
## haut        -0.6610221  0.2335215  -2.831  0.00866 **
## diam         0.1478040  0.0540151   2.736  0.01085 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5458 on 27 degrees of freedom
## Multiple R-squared:  0.6133,Adjusted R-squared:  0.5417
## F-statistic: 8.564 on 5 and 27 DF,  p-value: 5.87e-05
```

On observe que les p-values des tests t associés aux variables alt, pente Haut et diam sont significatifs.
Le coefficient de corrélation au carré vaut 0.61. 61% de la variation de proc est expliqué par la régression.

4.4 Recherche du meilleur modèle

Exemple R - 12 :

Utilisation de la fonction stepAIC.

```
library(MASS)
lm0<-lm(pin$proce~1)
?stepAIC
stepAIC(lm0,
        .~pin$alt+pin$pente+pin$densi+pin$haut+pin$diam,
        trace=F)

##
## Call:
## lm(formula = pin$proce ~ pin$densi + pin$pente + pin$alt)
##
## Coefficients:
## (Intercept)    pin$densi    pin$pente      pin$alt
##    4.945074    -0.021791    -0.036632    -0.002153
```

Le meilleur modèle retenu avec le critère d'AIC est le modèle avec densi, pente et alt. Regardons ce modèle de plus près.

Exemple R - 13 :

Utilisation de la fonction stepAIC.

```
pin.lm1<-lm(pin$proce ~ pin$densi + pin$pente + pin$alt)
summary(pin.lm1)

##
## Call:
## lm(formula = pin$proce ~ pin$densi + pin$pente + pin$alt)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.08006 -0.40544  0.02102  0.48947  1.40411
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.9450738  1.2989000   3.807 0.000674 ***
## pin$densi   -0.0217906  0.0140054  -1.556 0.130585
## pin$pente   -0.0366323  0.0148919  -2.460 0.020098 *
## pin$alt     -0.0021530  0.0009792  -2.199 0.036023 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6014 on 29 degrees of freedom
## Multiple R-squared:  0.4957,Adjusted R-squared:  0.4436
## F-statistic: 9.503 on 3 and 29 DF,  p-value: 0.0001563
```

Donc il semble que seul densi et alt sont vraiment significative dans le modèle. Nous allons étudier plus en détail les résidus du modèle retenu : *proce pente + alt*

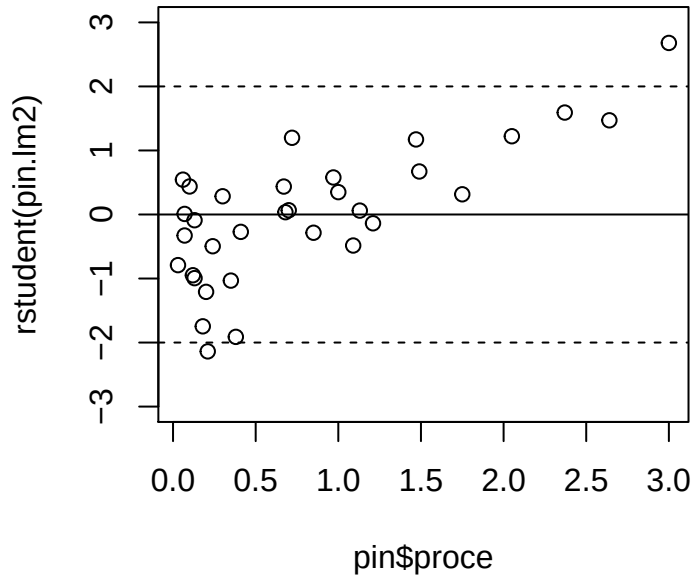
4.5 Etude des résidus

Exemple R - 14 :

Utilisation de la fonction stepAIC.

```
pin.lm2<-lm(pin$proce ~ pin$pente + pin$alt)
pin.lm2

##
## Call:
## lm(formula = pin$proce ~ pin$pente + pin$alt)
##
## Coefficients:
## (Intercept)    pin$pente      pin$alt
##    5.973055    -0.044278    -0.002957

residus<-pin.lm2$residuals
shapiro.test(residus)

##
##  Shapiro-Wilk normality test
##
## data:  residus
## W = 0.98587, p-value = 0.9352
```

```
plot(x=pin$proce,
     y=rstudent(pin.lm2),ylim=c(-3,3))
abline(+2,0,lty=2)
abline(0,0,lty=1)
abline(-2,0,lty=2)
```



Exercice à rédiger

Reprendre les données du modèle *modele*1 pour prédire le poids des arbres à partir de la variable *ddh*

On va donc chercher un autre modèle qui permettrait d'éviter ce problème de non normalité des résidus. Une transformation ln pourrait peut être améliorer le modèle.

Refaite l'analyse complète avec le modèle

$$\log(poids) = a.\log(ddh) + b \tag{1}$$

Rédiger un compte rendu (1 page recto-verso par binome sur cette analyse)