TRAVAUX PRATIQUES POUR L'APPRENTISSAGE DE L'UTILISATION DE R[1]

EXEMPLE DE L'AJUSTEMENT SELON LES MOINDRES CARRÉS

OBJECTIFS PÉDAGOGIQUES :

L'ajustement est un domaine dans lequel on peut expérimenter (conjecturer et mettre en œuvre les vérifications nécessaires), on peut réinvestir les résultats de l'analyse mathématique et la connaissance des fonctions et de leurs graphes, on peut entraîner les élèves à l'analyse de modèles, de leurs résultats et de leurs applications concrètes.

L'outil R est téléchargeable gratuitement (**R** : A Programming Environment for Data Analysis and Graphics, 1999–2004 R Development Core Team from the R-project.(www.r-project.org), il a les avantages et les inconvénients d'un outil professionnel, mais il est complet et relativement facile d'utilisation. Il possède des fonctionnalités graphiques étendues. Il est de plus une bonne source bibliographique sur les méthodes et algorithmes statistiques : Toutes les commandes font l'objet d'une description détaillée, dans laquelle figurent des exemples (parfois succincts) et une bibliographie précise.

L'exemple et les idées directrices de ce topo sont tirés des pages de préparation de deux articles (l'un de Stephan Manganelli (vol.I) et l'autre d'Hubert Raymondaud (vol.II) d'un ouvrage édité par l'A.P.M.E.P. (Statistiques au Lycée) en deux volumes, à paraître en octobre 2004 pour le volume I, (pour les journées nationales de l'A.P.M.E.P. : Association des Professeurs de Mathématiques de l'Enseignement Public) et dans le courant de l'année scolaire 2004-2005 pour le volume 2.

A) T.P.N°1 DÉTAILLÉ ET COMMENTÉ : COMPARAISON DE CINQ MODÈLES D'AJUSTEMENT D'UN NUAGE "NON LINÉAIRE"

(EXEMPLE N°4 DE L'ARTICLE DE H.RAYMONDAUD)

Il s'agit de trouver un modèle déterministe (simple description statistique non probabiliste) pour modéliser la relation entre l'âge (en année) d'une machine outil et son coût horaire d'entretien (en euro), à partir des données suivantes :

X : Âge (année)	1	2	3	4	5
Y : Coût horaire (euro)	13,3	14,2	16,1	18,9	23,6

Le tableau **(T0)** suivant résume quelques **stratégies** de modélisation et leurs résultats.

SÉRIE DOUBLE UTILISÉE : S.C.E. de Y ≈ 69,268		(X ; Y)	(X ; ln(Y))	(X ; Y^{-2})
MODÈLE AJUSTÉ	VARIABLE FINALE ESTIMÉE	f est l'identité	f est la fonction ln	f est la fonction puissance −2
$\{f(Y)\}_{estimé} = aX + b.$ par les moindres carrés linéaire **lsfit( )**.	f(Y). $SCE_{résiduelles}$ **non** comparables.	a ≈ 2,53 b ≈ 9,63 $SCE_{résiduelle}$ ≈5,259 (T1)	a ≈ 0,1432885 b ≈ 2,3941812 $SCE_{résiduelle}$ ≈ 0,0086 (T2)	a ≈ −0,00098754 b ≈ 0,0067756695 $SCE_{résiduelle}$ ≈$4,789x10^{-8}$
	$f^{-1}(\{f(Y)\}_{estimé})$. $SCE_{résiduelles}$ comparables.	$SCE_{résiduelle}$ ≈ 5,259	$SCE_{résiduelle}$ ≈ 2,7862 (T2)	$SCE_{résiduelle}$ ≈ 0,1705
$Y_{estimé} = f^{-1}(aX + b)$. par les moindres carrés non linéaire **nls( )**.	Y $SCE_{résiduelles}$ comparables.	a ≈ 2,53 b ≈ 9,63 $SCE_{résiduelle}$ ≈ 5,259	a ≈ 0,1537502 b ≈ 2,3605093 $SCE_{résiduelle}$ ≈ 2,4734 (T3)	a ≈ −0,001012942 b ≈ 0,006861522 $SCE_{résiduelle}$ ≈ 0,0895 (T4)

[1] Je remercie chaleureusement Claude BRUCHOU, du service biométrie de l'INRA d'Avignon, qui a bien voulu m'initier à R, et est toujours disponible pour répondre à mes interrogations statistiques.

$Y_{estimé} = a+bX+cX^2+dX^3$ par les moindres carrés multiple **lsfit( )**.	Y	a ≈ 12,62 b ≈ 0,65714286 c ≈ — 0,06785714 d ≈ 0,075 $SCE_{résiduelle}$ ≈ 0,01728571 (T6)		

L'ajustement d'un modèle polynômial relève de la méthode des moindres carrés multiple (cf. les commentaires de T6), mais on peut aussi utiliser un algorithme des moindres carrés non linéaire (cf. les commentaires de T5)

Notez que c'est le modèle polynomial qui a obtenu la meilleure performance quant à la SCE résiduelle, et il peut être intéressant d'expérimenter des polynômes de degré supérieur, on diminue alors encore la SCE résiduelle. Mais le modèle polynomial n'a pas que des avantages (cf.T5).

En effet, le critère de la SCE résiduelle minimale, s'il est pertinent pour le calcul des paramètres d'un modèle, ne l'est pas forcément pour le choix d'un modèle parmi plusieurs. Il est parfois préférable d'avoir un modèle dont les paramètres ont une interprétation concrète, même si sa résiduelle est plus élevée que, par exemple, un modèle polynômial dont les paramètres n'ont, en général, pas d'interprétation pratique simple et utilisable concrètement.

Le tableau T0 permet de bien montrer que la stratégie de la transformation n'est pas optimale quant au critère de la SCE résiduelle minimale concernant Y. La seule méthode optimale est les moindres carrés non linéaire.

Pour les lecteurs qui désirent élargir et/ou approfondir la question, les ouvrages d'initiation (1), (2) et (3), rédigés par des biométriciens et des agronomes, sont de très bonne facture.

Les 6 tableaux suivants présentent :

* Dans la colonne de droite, les **commandes R permettant de mettre en œuvre les méthodes statistiques nécessaires, et leurs résultats.** J'ai rajouté un certain nombre de commandes qui n'étaient pas forcément nécessaires à la mise en œuvre des méthodes statistiques, mais qui permettent de mieux comprendre le fonctionnement de R, la nature des objets qu'il utilise, les manipulations de données et de fichiers.
* Dans la colonne de gauche mes **commentaires sur les commandes R et sur les méthodes**.

Ils sont suivis par les graphiques correspondant et **leurs commentaires.**

Je me suis contenté d'évaluer deux critères de qualité des modèles ajustés :

* Un critère "qualitatif" qui est l'adéquation de la forme de la courbe avec la forme du nuage, qui se manifeste soit directement sur le nuage de point lui-même, soit sur le nuage des résidus par la présence plus ou moins marquée de "tendance" dans la répartition des points de chaque coté de la droite y=0, le cas le plus favorable étant celui où les points illustrant les résidus se répartissent de façon "homogène" de chaque coté de la droite y=0. Afin de préciser cette notion de tendance, on peut parler d'autocorrélation, bien que ce terme soit plus souvent employé pour les séries temporelles.
* Un critère quantitatif, qui est la **S**omme des **C**arrés de **É**cart résiduelle (S.C.E. r.) que l'on compare avec la S.C.E. totale de Y. Selon les modèles ou la stratégie d'ajustement utilisés, les S.C.E. r. ne sont pas toujours comparables (cf. T0).

T1 : Modèle "**cout=a*age+b**" ajusté par la méthode des moindres carrés linéaire, mise en œuvre par la commande "**lsfit**" de R(cf. biblio en fin de document).

COMMANDES R ET AFFICHAGES RÉSULTATS R	COMMENTAIRES

`> age=c(1,2,3,4,5)`
`> cout=c(13.3,14.2,16.1,18.9,23.6)`
 	
`> age`
`[1] 1 2 3 4 5`
`> cout`
`[1] 13.3 14.2 16.1 18.9 23.6`
	Saisie des données dans les "objets" age et cout, qui sont des vecteurs. "**c**" est une commande de R.
	La commande "**read.table( )**" permet de récupérer un tableau de données à partir d'un fichier texte
	Affichage du contenu des deux vecteurs.

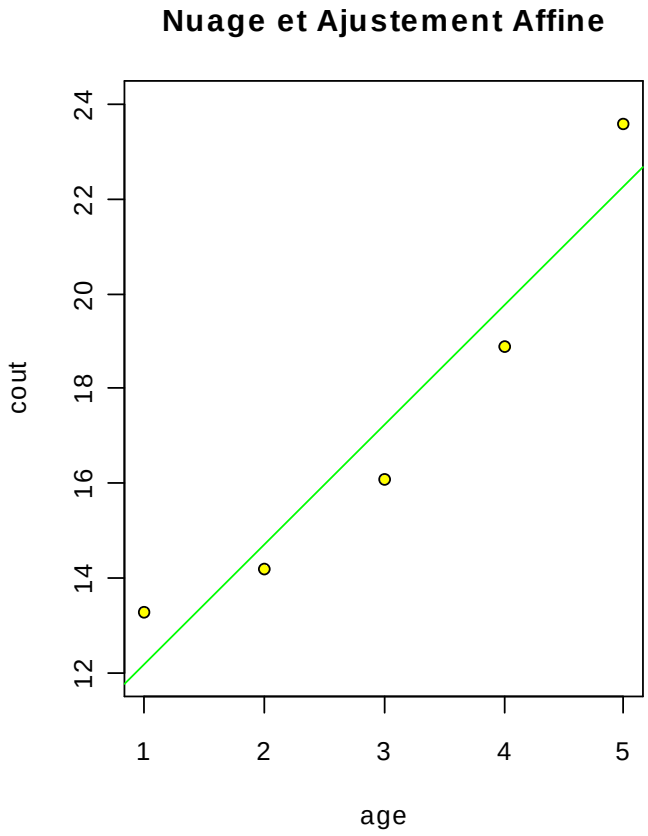
`> MAf=lsfit(age,cout)`
 	
`> names(MAf)`
`[1] "coefficients" "residuals" "intercept" "qr"`
 	
`> MAf$coefficients`
`Intercept X`
`9.63 2.53`
`> MAf$residuals`
`[1] 1.14 -0.49 -1.12 -0.85 1.32`
	Le modèle ajusté est **cout=a*age+b.**
	Utilisation de la commande R "**lsfit**" : moindres carrés linéaire.
	La commande "**names**" affiche les éléments de l'objet R, résultat de la commande "**lsfit**".
	
	Affichage des paramètres du modèle.
	Affichage des résidus du modèle.

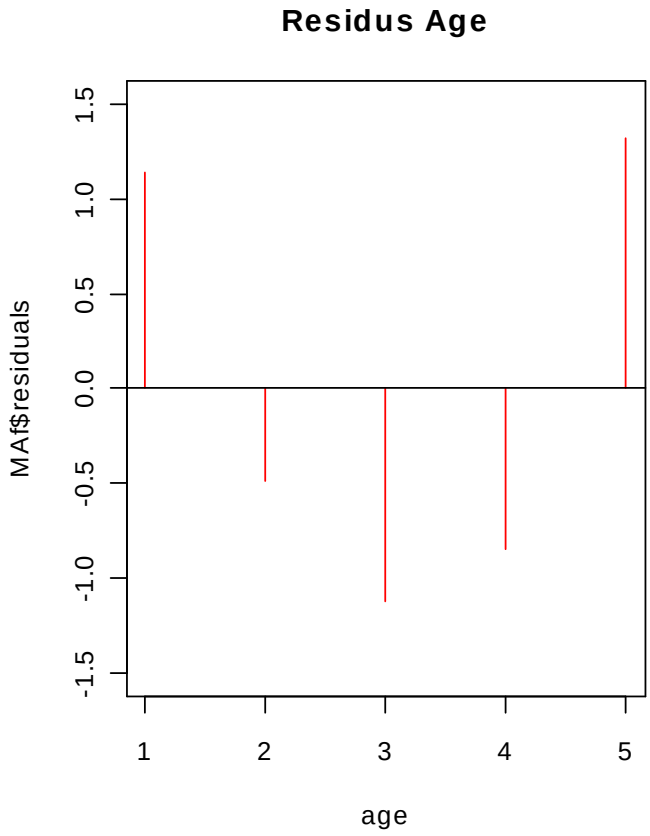
`> MAfEst=MAf$coefficients[1]+MAf$coefficients[2]*age`
`> MAfEst`
`[1] 12.16 14.69 17.22 19.75 22.28`
 	
`>`
`SCEMAf=sum((MAf$residuals)^2);SCETotaleY=var(cout)*4`
`> SCEMAf;SCETotaleY`
`[1] 5.259`
`[1] 69.268`
	Calcul puis affichage des valeurs de Y estimées par le modèle.
	
	Calcul puis affichage de la S.C.E. résiduelle en Y et de la S.C.E.totale de Y.

`> ls()` `[1] "age" "cout" "MAf" "MAfEst" "SCEMAf"`	Affichage des objets R contenus dans la mémoire de travail.

`> par(mfrow=c(1,2),ask=T)`
 	
`> plot(age,cout,pch=21,bg="yellow",ylim=c(12,24));`
`abline(MAf$coefficients,col="green");title("Nuage et Ajustement Affine")`
`Hit <Return> to see next plot:`
`> plot(age,MAf$residuals,type="h",col="red",`
`ylim=c(-1.5,1.5));abline(h=0);title("Residus Age")`
 	
`{> plot(MAf$residuals,type="h",col="red",`
`ylim=c(-1.5,1.5));abline(h=0);title("Residus Index")`
 	
`> plot(cout,MAf$residuals,type="h",col="red",`
`xlim=c(12,24),ylim=c(-1.5,1.5));abline(h=0);`
`title("Résidus Coût")}`
	Découpage de la fenêtre graphique en 1 ligne et 2 colonnes.
	Construction/affichage du nuage de point, de la courbe représentative du modèle ajusté et de son titre.
	
	Construction/affichage d'un "nuage en bâtons" des résidus, de la droite y=0 et du titre.
	
	Autres représentations possibles des résidus, avec diverses abscisses.

`> save.image("D:/HubW/MATH/FiR/RDidactAgeCout.RData")`	Enregistrement sur fichier des objets contenus dans la mémoire de travail.





Le nuage de points, comme le "nuage" des résidus montrent l'inadéquation du modèle "**cout=a*age+b**" à "bien" représenter la relation entre X et Y. L'autocorrélation (la "tendance") des résidus est trop marquée, la S.C.E. résiduelle est importante (5,259 ; 7,59% de la totale) par rapport à la S.C.E. totale des y (69.268).

C'est pour cela que l'énoncé original propose d'utiliser la transformation logarithme népérien de Y. C'est ce qui est fait dans le tableau suivant.

T2 : Modèle "ln(**cout)=a*age+b**" ajusté par la méthode des moindres carrés linéaire, mise en œuvre par la commande **"lsfit"** de R.

COMMANDES R ET AFFICHAGES RÉSULTATS R	COMMENTAIRES

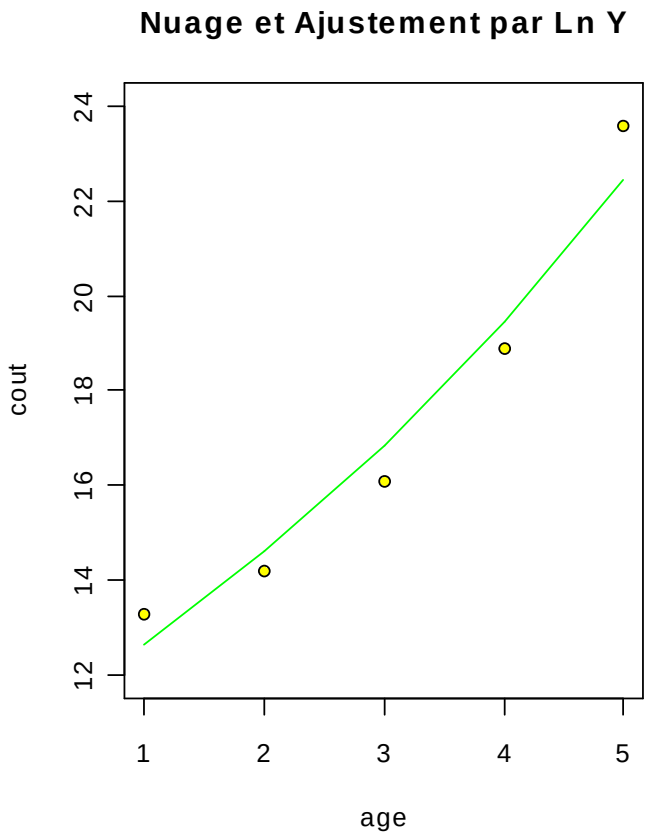
`load("D:/HubW/MATH/FiR/RDidactAgeCout.RData")`
 	
`> ls()`
`[1] "age" "cout"`
`[3] "last.warning" "MAf"`
`[5] "MAfEst" "SCEMAf"`
	Chargement dans la mémoire de travail des objets contenus dans le fichier enregistré à la sessions précédente.
	Affichage des objets R contenus dans la mémoire de travail.

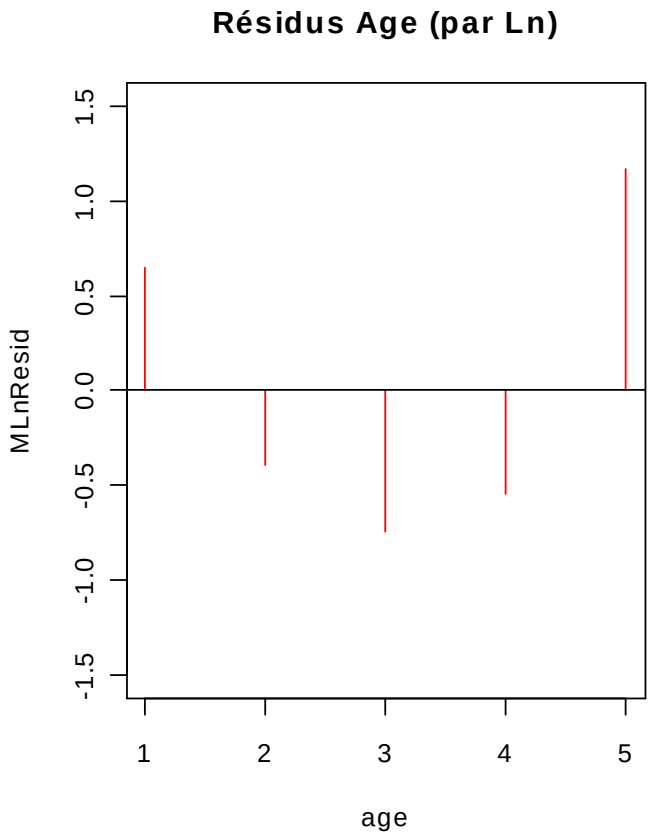
`> MLn=lsfit(age,log(cout))`
 	
`> MLn$coefficients`
`Intercept X`
`2.3941812 0.1432885`
 	
`> MLnEst=exp(MLn$coefficients[1]+MLn$coefficients[2]*age)`
`> MLnEst`
`[1] 12.64763 14.59616 16.84488 19.44005 22.43504`
	Le modèle ajusté est **ln(cout)=a*age+b.**
	Utilisation de l'algorithme des moindres carrés linéaire sur la série transformée (X,ln(Y)).
	Affichage des paramètres du modèle.
	Calcul puis affichage des valeurs estimées de Y.

`> MLnResid=cout-MLnEst`
`> MLnResid`
`[1] 0.6523716 -0.3961566 -0.7448805 -0.5400489`
`1.1649640`
 	
`> SCELn=sum(MLnResid^2)`
`> SCELn`
`[1] 2.786170`
	Calcul puis affichage des résidus en Y.
	
	Calcul puis affichage de la S.C.E. résiduelle en Y.

`> par(mfrow=c(1,2),ask=T)`
 	
`> plot(age,cout,pch=21,bg="yellow",ylim=c(12,24));`
`lines(age,MLnEst,col="green");title("Nuage et Ajustement par Ln Y")`
`Hit <Return> to see next plot:`
 	
`> plot(age,MLnResid,type="h",col="red",ylim=c(-1.5,1.5));`
`abline(h=0);title("Résidus Age (par Ln)")`
	Graphiques.

`> ls()`
`[1] "age" "cout" "last.warning" "MAf"`
`"MAfEst" "MLn"`
`[7] "MLnEst" "MLnResid" "SCELn"`
`"SCEMAf"`
 	
`> save.image("D:/HubW/MATH/FiR/RDidactAgeCout.RData")`
	Affichage des objets R contenus dans la mémoire de travail.
	
	Enregistrement des objets dans un fichier.





Bien qu'il y ait une petite amélioration quantitative sur les résidus (S.C.E. résiduelle plus petite : 2,786 ; 4,02% de la totale), par rapport au modèle **"cout=a*age+b"**, le modèle **"ln(cout)=a*age+b"** ne représente pas la relation entre X et Y de façon satisfaisante. "Qualitativement", l'autocorrélation (la "tendance") entre les résidus reste trop marquée.

Je propose donc d'utiliser un algorithme permettant d'ajuster directement le modèle exponentiel qui découle de la transformation ln(Y). C'est ce qui est fait dans le tableau suivant.

T3 : Modèle "cout=e^(a*age+b)" ajusté par l'algorithme des moindres carrés non linéaire "nls" de R(cf. biblio en fin de document).

COMMANDES R ET AFFICHAGES RÉSULTATS R	COMMENTAIRES

`> MExpo=nls(cout~exp(a*age+b),start=list(a=.1,b=2.4))`
`> MExpo`
`Nonlinear regression model`
`  model:  cout ~ exp(a * age + b)`
`   data:  parent.frame()`
`        a         b`
`0.1537502 2.3605093`
` residual sum-of-squares:  2.473443`
	Le modèle ajusté est **cout=e^(a*age+b)**.
	Utilisation de l'algorithme des moindres carrés non linéaire (**nls**) sur la série (X,Y)).
	Affichage des paramètres du modèle et de la S.C.E. résiduelle en Y.
	Cf. REMARQUE du T4.

`> SuMExpo=summary(MExpo)` `> SuMExpo` `Formula: cout ~ exp(a * age + b)` `Parameters:` `  Estimate Std. Error t value Pr(>|t|)` `a  0.15375    0.01731   8.881  0.00301 **` `b  2.36051    0.06631  35.597 4.88e-05 ***` `Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1` `Residual standard error: 0.908 on 3 degrees of freedom` `Correlation of Parameter Estimates:` ` a` `b -0.9375`	La commande "**summary**" permet de calculer et d'afficher des résultats supplémentaires obtenus à partir de l'objet Mexpo.

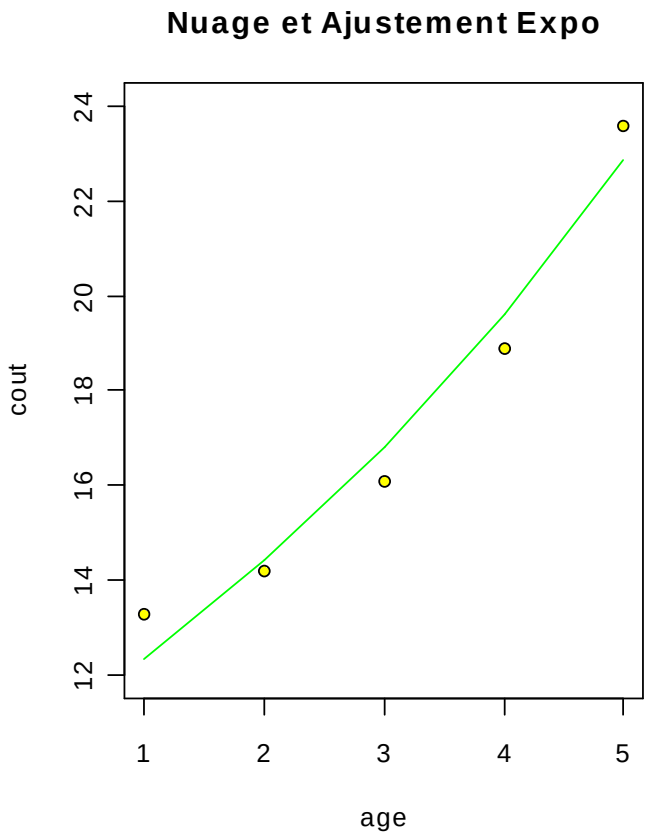
`> names(SuMExpo)`
`[1] "formula"  "residuals"  "sigma"  "df"`
`"cov.unscaled" "correlation"`
`[7] "parameters"`
	La commande "**names**" affiche les éléments de l'objet R SuMExpo, résultat de la commande "**nls**".

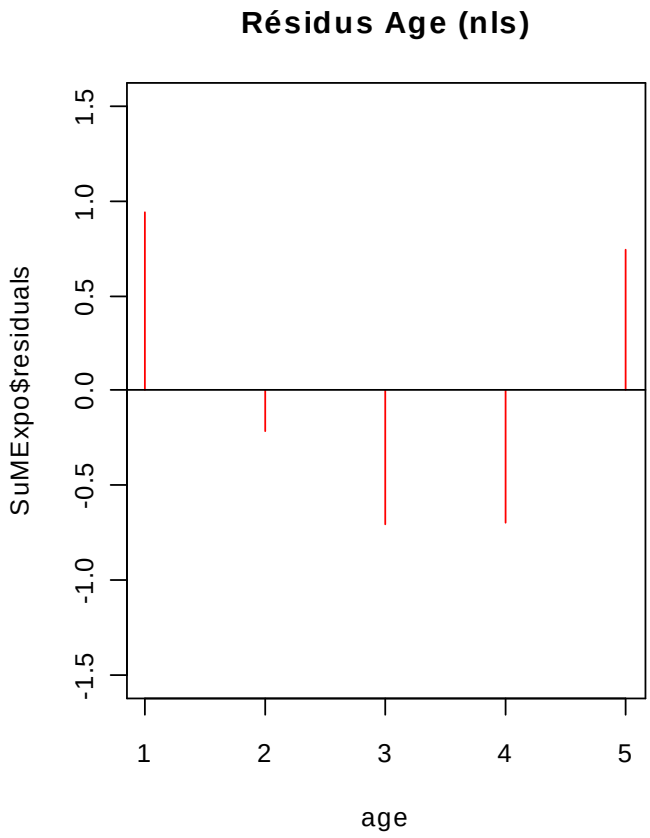
`> SuMExpo$residuals`
`[1]  0.9425456 -0.2112575 -0.7064016 -0.6996175`
`0.7429355`
	Affichage des résidus en Y.

`> MExpoEst=predict(MExpo)`
`> MExpoEst`
`[1] 12.35745 14.41126 16.80640 19.59962 22.85706`
	Calcul puis affichage des valeurs estimées de Y.

`> par(mfrow=c(1,2),ask=T)`
 	
`> plot(age,cout,pch=21,bg="yellow",ylim=c(12,24));`
`lines(age,MExpoEst,col="green");title("Nuage et`
`Ajustement Expo")`
`Hit <Return> to see next plot:`
`> plot(age,SuMExpo$residuals,type="h",col="red",`
`ylim=c(-1.5,1.5));abline(h=0);title("Résidus Age (nls)")`
	Graphiques.

`> ls()`
` [1] "age"  "cout"  "last.warning" "MAf"`
`"MAfEst"  "MExpo"`
` [7] "MExpoEst"  "MLn"  "MLnEst"`
`"MLnResid"  "SCELn"  "SCEMAf"`
`[13] "SuMExpo"`
	Affichage des objets R contenus dans la mémoire de travail.



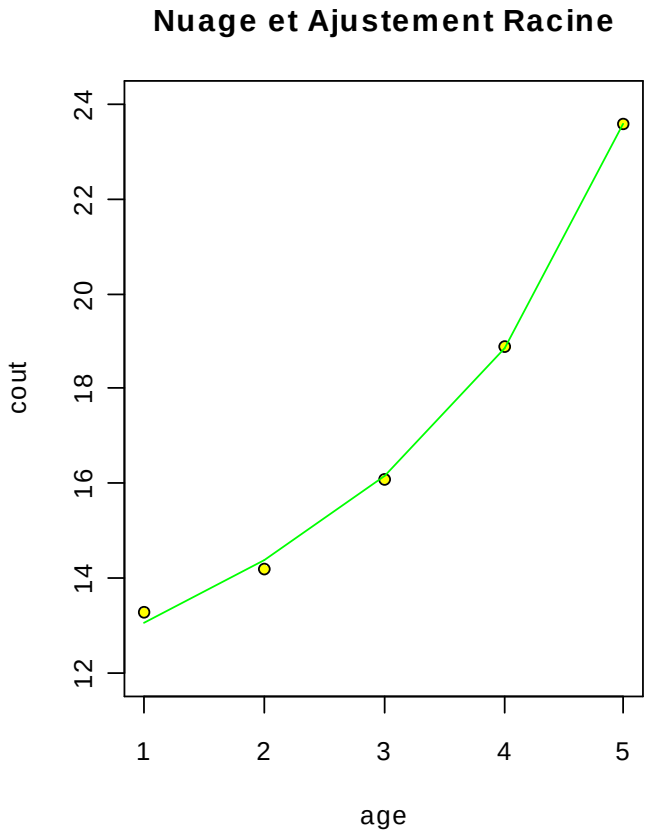


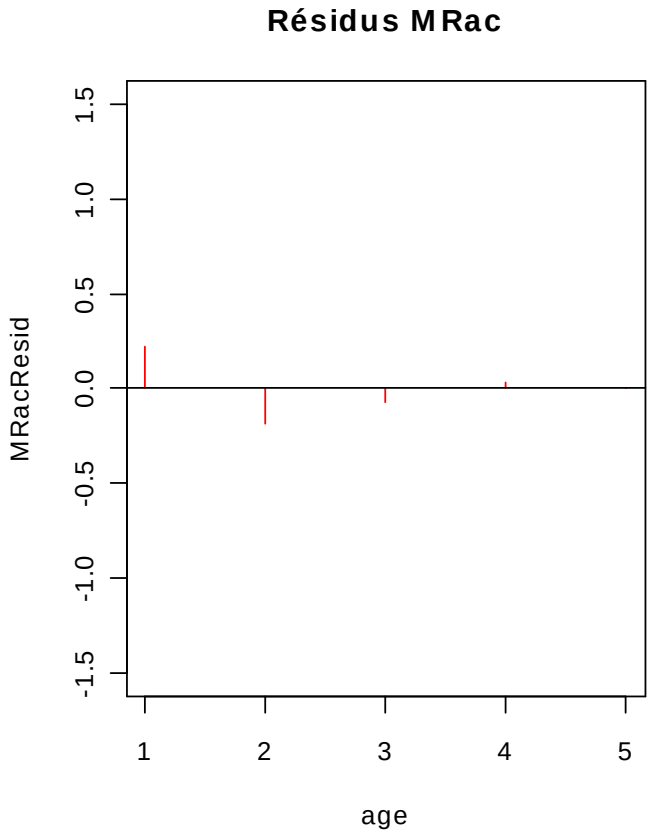
Une très légère amélioration quantitative, la S.C.E. résiduelle (2,473 ; 3,57% de la totale) a encore un peu diminué par rapport aux deux modèles précédents, mais le nuage de points avec la courbe ajustée, comme le "nuage" des résidus montrent que le modèle "**cout=e$^{(a*age+b)}$**" n'est pas satisfaisant. L'autocorrélation reste trop marquée.

Je propose donc d'utiliser l'échelle des transformations de Tukey (cf. biblio en fin de document) permettant de rechercher une transformation s'adaptant le mieux possible au nuage à étudier. C'est ce qui est fait dans le tableau suivant.

T4 : Modèle "**cout = (a*age+b)$^{-1/2}$**" ajusté par l'algorithme des moindres carrés non linéaire "**nls**" de R.

COMMANDES R ET AFFICHAGES RÉSULTATS R	COMMENTAIRES
`> MRac=nls(cout~1/sqrt(a*age+b),start=list(a=-0.001,b=0.007))` `> MRac` `Nonlinear regression model` `  model:  cout ~ 1/sqrt(a * age + b)` `   data:  parent.frame()` `           a           b` `-0.001012943  0.006861523` ` residual sum-of-squares:  0.08948713` `> summary(MRac)` `Formula: cout ~ 1/sqrt(a * age + b)` `Parameters:` `    Estimate Std. Error t value Pr(>|t|)` `a -1.013e-03  2.377e-05  -42.61 2.85e-05 ***` `b  6.862e-03  1.089e-04   62.99 8.82e-06 ***` `---` `Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1` `Residual standard error: 0.1727 on 3 degrees of freedom` `Correlation of Parameter Estimates:` ` a` `b -0.9796`  `> MRacEst=predict(MRac)` `> MRacEst` `[1] 13.07600 14.38047 16.17392 18.86540 23.59114`  `> MRacResid=(summary(MRac))$residuals` `> MRacResid` `[1] 0.224004240 -0.180471563 -0.073916004 0.034601023 0.008856526`  `> par(mfrow=c(1,2),ask=T)` `> plot(age,cout,pch=21,bg="yellow",ylim=c(12,24));` ` lines(age,MRacEst,col="green");` ` title("Nuage et Ajustement Racine")` `Hit <Return> to see next plot:` `> plot(age,MRacResid,type="h",col="red",` ` ylim=c(-1.5,1.5));abline(h=0);title("Résidus MRac")`  `***> MRac=nls(cout~1/sqrt(a*age+b),start=list(a=1,b=1))` `Error in numericDeriv(form[[3]], names(ind), env) :` ` Missing value or an Infinity produced when` ` evaluating the model` `In addition: Warning message:` `NaNs produced in: sqrt(a * age + b)`	Le modèle ajusté est **cout = (a*age+b)$^{-1/2}$.** Utilisation de l'algorithme des moindres carrés non linéaire (**nls**) sur la série (X,Y)). Affichage des paramètres du modèle et de la S.C.E. résiduelle en Y. **REMARQUE sur l'algorithme nls, qui vaut aussi pour les algorithmes opérant par approximation successives (par exemple le Marquardt qui est utilisé dans la procédure HAUSS59 de la logithèque statistique des laboratoires de biométrie de l'INRA, cf. biblio.) :** L'algorithme nécessite qu'on lui fournisse des valeurs de "départ" pour les paramètres du modèle (c'est l'objet de la commande "start" dans **nls**). Il est donc indispensable d'avoir une idée pas trop fausse de la valeur de ces paramètres. Ce qui implique qu'il faut connaître une façon d'en approcher les valeurs. Avec des valeurs de départ trop éloignées, l'algorithme peut ne pas fonctionner, s'il rencontre, lors des itérations qu'il effectue, des problèmes pour calculer les valeurs des fonctions qu'il utilise. C'est ce qui se passe dans l'exemple *** en bleu à gauche. Si l'on propose comme valeurs de départ a=1 et b=1, l'algorithme ne peut pas terminer ses itérations et il indique un message d'erreur. C'est une des principales difficultés de l'utilisation de tels algorithmes. C'est aussi ce qui les rend, à mon avis, pédagogiquement intéressants. Le calcul n'est plus automatique comme dans les moindres carrés linéaire.





Il y a une nette amélioration tant qualitative (l'autocorrélation est moins marquée) que quantitative (la S.C.E.résiduelle de 0,089 (0,129% de la totale) est nettement plus faible que dans les autres modèles. La méthode de l'échelle de Tukey (cf. biblio en fin de document) a bien porté ses fruits.

Mais on peut encore "mieux" faire : on peut obtenir une plus petite S.C.E. résiduelle, et une meilleure conformation des résidus. C'est ce que l'on va voir dans le tableau suivant.

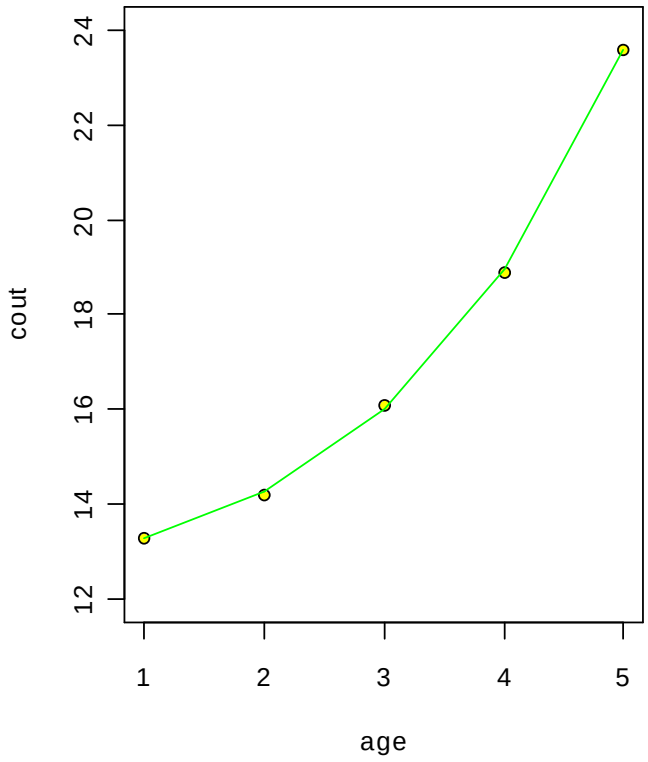
T5 : Modèle "cout = a+b*age+c*age²+d*age³" ajusté par l'algorithme des moindres carrés non linéaire "nls" de R.

COMMANDES R ET AFFICHAGES RÉSULTATS R	COMMENTAIRES
`> MPol3=nls(cout~a+b*age+c*age^2+d*age^3,` `  start=list(a=1,b=1,c=1,d=1))` `> MPol3` `Nonlinear regression model` `  model:  cout ~ a + b * age + c * age^2 + d * age^3` `   data:  parent.frame()` `          a           b           c           d` `12.62000000  0.65714286 -0.06785714  0.07500000` ` residual sum-of-squares:  0.01728571` `> summary(MPol3)` `Formula: cout ~ a + b * age + c * age^2 + d * age^3` `Parameters:` `  Estimate Std. Error t value Pr(>|t|)` `a 12.62000    0.64677  19.512   0.0326 *` `b  0.65714    0.84543   0.777   0.5794` `c -0.06786    0.31379  -0.216   0.8644` `d  0.07500    0.03465   2.165   0.2755` `---` `Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1` `  ` ' 1` `Residual standard error: 0.1315 on 1 degrees of freedom` `Correlation of Parameter Estimates:` ` a b c` `b -0.9747` `c 0.9369 -0.9890` `d -0.9000 0.9672 -0.9937`	Le modèle ajusté est **cout=a+b*age+c*age²+d*age³.** **REMARQUES sur le choix d'un modèle :** Pour ajuster ce genre de nuage ("sans épaisseur") de façon quasi "parfaite", il suffit de prendre comme modèle un polynôme de degré suffisamment élevé. Il se pose alors le problème de la signification et de l'interprétation pratique des valeurs des paramètres. Ils n'ont, en général, plus de signification pratique simple. C'est l'inconvénient majeur des modèles polynômiaux. Un autre problème en découle qui est celui de l'évaluation des valeurs initiales des paramètres. Comme ils n'ont pas de signification pratique, comment peut-on évaluer des valeurs initiales?

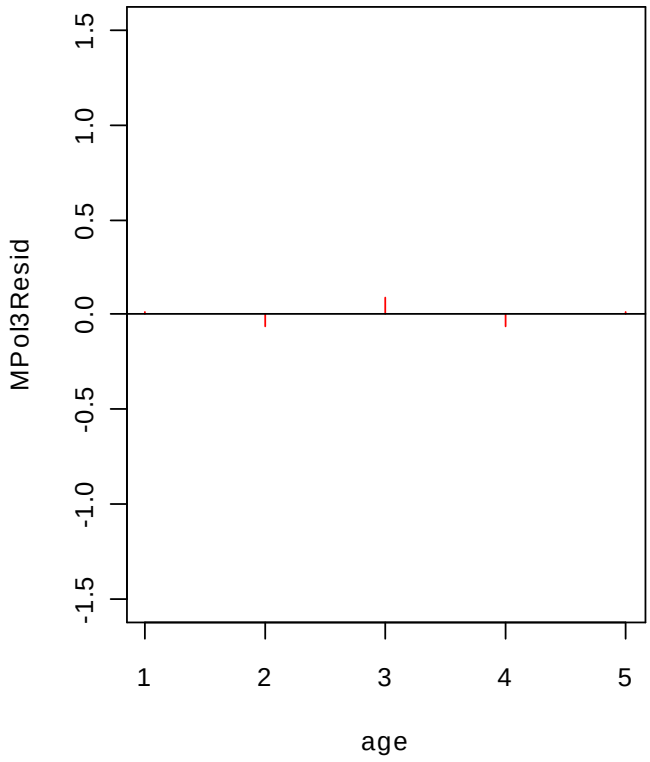
`> MPol3Est=predict(MPol3)`
`> MPol3Est`
`[1] 13.28429 14.26286 16.00571 18.96286 23.58429`
 	
`> MPol3Resid=(summary(MPol3))$residuals`
`> MPol3Resid`
`[1]  0.01571429 -0.06285714  0.09428571 -0.06285714`
`  0.01571429`
	En fait, le modèle polynômial est un modèle linéaire, car les paramètres interviennent de façon linéaire dans le modèle. Mais c'est un modèle linéaire à plusieurs variables qui sont x, x², x³ ...
	Pas de problème, la commande **"lsfit"** de R sait faire.
	C'est ce que nous allons voir dans le tableau suivant T6.

`> par(mfrow=c(1,2),ask=T)`
`> plot(age,cout,pch=21,bg="yellow",ylim=c(12,24));`
`  lines(age,MPol3Est,col="green");`
`  title("Nuage et Ajustement Poly3.")`
`Hit <Return> to see next plot:`
`> plot(age,MPol3Resid,type="h",col="red",`
`  ylim=c(-1.5,1.5));abline(h=0);title("Résidus MPol3")`
	Graphiques.

Nuage et Ajustement Poly3.



Résidus MPol3



L'ajustement par le modèle **"cout = a+b*age+c*age²+d*age³"** est satisfaisant, tant du point de vue quantitatif—la S.C.E. résiduelle (0,017 ; 0,02% de la totale) est la plus petite de toutes celles déjà obtenues, que du point de vue "qualitatif"—l'autocorrélation est la moins marquée de toutes.

Est-ce pour autant le modèle que l'on choisirait dans la pratique ? Rien n'est moins sûr. Vous pourrez poursuivre cette lecture dans les articles de Stephan Manganelli et Hubert RAYMONDAUD (dans "statistiques au lycée", volumes I et II, publié par l'APMEP, cf. biblio.).

T6 : Modèle **"cout = a+b*age+c*age²+d*age³"** ajusté par la méthode des moindres carrés linéaire, mise en œuvre par la commande **"lsfit"** de R.

COMMANDES R ET AFFICHAGES RÉSULTATS R	COMMENTAIRES

`> age3=cbind(age,age^2,age^3)`
`> age3`
`     age`
`[1,]   1  1   1`
`[2,]   2  4   8`
`[3,]   3  9  27`
`[4,]   4 16  64`
`[5,]   5 25 125`
`> MPol3Lin=lsfit(age3,cout)`
`> MPol3Lin$coefficients`
`  Intercept          age`
`12.62000000  0.65714286 -0.06785714  0.07500000`
`> MPol3Lin$residuals`
`[1]  0.01571429 -0.06285714  0.09428571 -0.06285714`
`  0.01571429`
`> MPol3LinSCEResid=sum(MPol3Lin$residuals^2)`
`> MPol3LinSCEResid`
`[1] 0.01728571`
	Le modèle ajusté est **cout=a+b*age+c*age²+d*age³**
	La commande **"cbind"** permet de construire une matrice **age3** dont les 3 colonnes sont les vecteurs age, age^2, age^3.
	
	L'ajustement **multiple** selon les moindres carrés linéaire est effectué par la commande **"lsfit"** de R, sous la forme matricielle **Y=X.A+e**. La méthode minimise le produit scalaire **e'e**. (cf. quelques détails en bas du tableau).
	
	On observe que l'on retrouve bien exactement les mêmes résultats qu'avec l'algorithme non linéaire.
	
	Lorsque l'on ajuste un modèle linéaire, les deux méthodes sont équivalentes.
	
	REMARQUES :
	À gauche, **trois** façons différentes de calculer les valeurs estimées de Y.

`> MPol3LinEst=cout-MPol3Lin$residuals`
`> MPol3LinEst`
`[1] 13.28429 14.26286 16.00571 18.96286 23.58429`
`>`
`  MPol3Lin$coefficients[1]+MPol3Lin$coefficients[2]*age`
`  +MPol3Lin$coefficients[3]*(age^2)+`
`  MPol3Lin$coefficients[4]*(age^3)`
`[1] 13.28429 14.26286 16.00571 18.96286 23.58429`	

`> age0=c(1,1,1,1,1)`
`> age3_1=cbind(age0,age3)`
`> age3_1`
`     age0 age`
`[1,]    1   1  1   1`
`[2,]    1   2  4   8`
`[3,]    1   3  9  27`
`[4,]    1   4 16  64`
`[5,]    1   5 25 125`
 	
`A=cbind(MPol3Lin$coefficients)`
`> A`
`                [,1]`
`Intercept 12.62000000`
`age        0.65714286`
`          -0.06785714`
`           0.07500000`
	`> YEstimé=age3_1%*%A`
	`> YEstimé`
	`           [,1]`
	`[1,] 13.28429`
	`[2,] 14.26286`
	`[3,] 16.00571`
	`[4,] 18.96286`
	`[5,] 23.58429`
	
	`> cout-YEstimé`
	`             cout`
	`[1,]  0.01571429`
	`[2,] -0.06285714`
	`[3,]  0.09428571`
	`[4,] -0.06285714`
	`[5,]  0.01571429`
		Détail de la forme matricielle **YEstimé=X.A : YEstimé** est la matrice colonne des coûts estimés, **X** est la matrice **"age3_1"** des variables explicatives à laquelle on a ajouté le vecteur colonne identité afin de prendre en compte le terme constant du polynôme; **A** est la matrice colonne des coefficients a, b, c, d.
		
		%*% est le produit matriciel.
		
		On peut obtenir e en faisant cout-YEstimé.

B) AUTRES TRAVAUX PRATIQUES NON COMMENTÉS

T.P.N°2 : UN PROBLÈME DE MAINTENANCE.

(PARAGRAPHE 5 DE L'ARTICLE DE S. MANGANELLI)

```
# Les commandes R sont précédées de >, les commentaires de #, s'il n'y a rien, ce sont les réponses affichées par R.
# Les variables sont : heures de fonctionnement (heures), pourcentage du parc en service (pcservi).
> rm(list=ls(all=TRUE)) # Toutes les variables sont vidées.
> heupa=read.table("C:/….../RegNl.txt",header=T,skip=31,nrows=9) # Lecture du fichier texte, les données sont tranférées dans la table "heupa".
> attach(heupa) # les données de travail par défaut sont dans la table "heupa".
```

```
# DONNÉES :
> heupa
  heures pcservi
1    100    0.80
2    200    0.64
3    300    0.52
4    400    0.40
5    500    0.32
6    600    0.28
7    750    0.20
8   1000    0.12
9   1500    0.04
```

```
# AJUSTEMENT NON LINÉAIRE :
> nlshp=nls(pcservi~exp(a*heures+b),start=list(a=-0.001,b=-0.03))
# start : il faut saisir des valeurs de départ pour les paramètres
> nlshp
Nonlinear regression model
  model:  pcservi ~ exp(a * heures + b)
   data:  parent.frame()
           a           b
-0.002173332 -0.010971682
 residual sum-of-squares:  0.0006711608

# AJUSTEMENT LINÉAIRE SUR CHANGEMENT DE VARIABLE LN(Y) :
> transhp=lsfit(heures,log(pcservi))
> transhp$coefficients
   Intercept            X
-0.029514716 -0.002118690

# calcul de la SCE résiduelle
> pred=exp(log(pcservi)-transhp$residuals)
> SCEr=sum((pcservi-pred)^2)
> SCEr
[1] 0.0008681539
```

```
# AJUSTEMENTS POLYNOMIAUX VIA L'ALGORITHME NON LINÉAIRE :
> poly2hp=nls(pcservi~a*heures^2+b*heures+c,start=list(a=1,b=1,c=1))
> poly2hp
Nonlinear regression model
  model:  pcservi ~ a * heures^2 + b * heures + c
   data:  parent.frame()
           a           b           c
 5.145209e-07 -1.324164e-03  8.860505e-01
 residual sum-of-squares:  0.007019438
> poly3hp=nls(pcservi~a*heures^3+b*heures^2+c*heures+d,start=list(a=1,b=1,c=1,d=1))
> poly3hp
Nonlinear regression model
  model:  pcservi ~ a * heures^3 + b * heures^2 + c * heures + d
   data:  parent.frame()
           a           b           c           d
-4.498055e-10  1.572124e-06 -1.972287e-03  9.781563e-01
 residual sum-of-squares:  0.0006422214
>poly4hp=nls(pcservi~a*heures^4+b*heures^3+c*heures^2+d*heures+e,start=list(a=1,b=1,c=1,d=1,e=1))
> poly4hp
Nonlinear regression model
  model:  pcservi ~ a * heures^4 + b * heures^3 + c * heures^2 + d * heures +      e
   data:  parent.frame()
           a           b           c           d           e
 3.217282e-13 -1.418966e-09  2.501739e-06 -2.290950e-03  1.007669e+00
 residual sum-of-squares:  0.0004223179
```

T.P.N°3 : ÉTUDE DE LA CROISSANCE D'UNE POPULATION MICROBIENNE

(ACTIVITÉS PLURIDISCIPLINAIRES, ANNEXE 2 DE L'ARTICLE DE S. MANGANELLI)

```
# Les variables sont : Heures (heures) et Croissance pondérale des bactéries (masse).
> heurbac=read.table("C:/…/RegNl.txt",header=T,skip=41,nrows=15)
> attach(heurbac)
```

```
# DONNÉES :
> heurbac
   heures masse msurm0
1       0  0.50   1.00
2       2  0.67   1.34
3       4  0.89   1.78
4       6  1.18   2.36
5       8  1.56   3.12
6      10  2.07   4.14
7      12  2.74   5.48
8      14  3.63   7.26
9      16  4.78   9.56
10     18  6.28  12.56
11     20  8.21  16.42
12     22 10.65  21.30
13     24 13.68  27.36
14     26 17.31  34.62
15     28 21.44  42.88

# AJUSTEMENT NON LINÉAIRE :
> nlshmas=nls(masse~exp(a*heures+b),
start=list(a=0.1,b=0.3))
>  nlshmas
Nonlinear regression model
  model:  masse ~ exp(a * heures +
b)
   data:  parent.frame()
         a          b
 0.1252982 -0.4206619
 residual sum-of-squares:  0.9009527
```

```
# AJUSTEMENT LINÉAIRE SUR CHANGEMENT DE VARIABLE
LN(Y) :
> transm=lsfit(heures,log(masse))
> transm
$coefficients
 Intercept          X
-0.6456694  0.1357858
> mpred=exp(log(masse)-transm$residuals)
> SCERe=sum((masse-mpred)^2)
> SCERe
[1] 4.78205


# AJUSTEMENT NON LINÉAIRE :
> nlshbac=nls(msurm0~exp(a*heures+b),start=list(
a=0.1,b=0.3))
> nlshbac
Nonlinear regression model
  model:  msurm0 ~ exp(a * heures + b)
   data:  parent.frame()
         a          b
0.1252982 0.2724850
 residual sum-of-squares:  3.603811

# AJUSTEMENT LINÉAIRE SUR CHANGEMENT DE VARIABLE
LN(Y) :
> transhbac=lsfit(heures,log(msurm0))
> transhbac$coefficients
 Intercept          X
0.04747774 0.13578583

# calcul de la SCE résiduelle
> bacpred=exp(log(msurm0)-transhbac$residuals)
> SCEResid=sum((bacpred-msurm0)^2)
> SCEResid
[1] 19.1282
```

```
# AJUSTEMENTS POLYNOMIAUX VIA L'ALGORITHME NON LINÉAIRE :
> poly2Mhbac=nls(masse~a*heures^2+b*heures+c,start=list(a=1,b=1,c=1))
> poly2Mhbac
Nonlinear regression model
  model:  masse ~ a * heures^2 + b * heures + c
   data:  parent.frame()
          a           b           c
 0.03759535 -0.37649111  1.46773529
 residual sum-of-squares:  5.312538

> poly3Mhbac=nls(masse~a*heures^3+b*heures^2+c*heures+d,start=list(a=1,b=1,c=1,d=1))
> poly3Mhbac
Nonlinear regression model
  model:  masse ~ a * heures^3 + b * heures^2 + c * heures + d
   data:  parent.frame()
           a            b           c            d
 0.001199121 -0.012767745  0.168389565  0.420183007
 residual sum-of-squares:  0.04104726
```

T.P.N°3 (SUITE)

```
> poly4Mhbac=nls(masse~a*heures^4+b*heures^3+c*heures^2+d*heures+e,start=list(a=1,b=1,c=1
,d=1,e=1))
> poly4Mhbac
Nonlinear regression model
  model:  masse ~ a * heures^4 + b * heures^3 + c * heures^2 + d * heures +      e
   data:  parent.frame()
           a            b            c            d            e
 9.477308e-06  6.683919e-04 -3.414995e-03  1.145585e-01  4.722248e-01
 residual sum-of-squares:  0.02356880

# AJUSTEMENTS POLYNOMIAUX VIA L'ALGORITHME NON LINÉAIRE :
> poly2hbac=nls(msurm0~a*heures^2+b*heures+c,start=list(a=1,b=1,c=1))
> poly2hbac
Nonlinear regression model
  model:  msurm0 ~ a * heures^2 + b * heures + c
   data:  parent.frame()
          a           b           c
 0.07519069 -0.75298222  2.93547059
 residual sum-of-squares:  21.25015

> poly3hbac=nls(msurm0~a*heures^3+b*heures^2+c*heures+d,start=list(a=1,b=1,c=1,d=1))
> poly3hbac
Nonlinear regression model
  model:  msurm0 ~ a * heures^3 + b * heures^2 + c * heures + d
   data:  parent.frame()
           a           b           c           d
 0.002398242 -0.025535490  0.336779130  0.840366013
 residual sum-of-squares:  0.1641890
>
poly4hbac=nls(msurm0~a*heures^4+b*heures^3+c*heures^2+d*heures+e,start=list(a=1,b=1,c=1,d
=1,e=1))
> poly4hbac
Nonlinear regression model
  model:  msurm0 ~ a * heures^4 + b * heures^3 + c * heures^2 + d * heures +      e
   data:  parent.frame()
           a            b            c            d            e
 1.895462e-05  1.336784e-03 -6.829991e-03  2.291169e-01  9.444496e-01
 residual sum-of-squares:  0.09427522

# AJUSTEMENTS POLYNOMIAUX VIA L'ALGORITHME LINÉAIRE, LES RÉGRESSEURS ÉTANT MULTIPLES
# (RÉGRESSION MULTIPLE DE Y SUR X, X^2, X^3, X^4 ...)
```

```
# GÉNÉRER LA MATRICE HEURES3 :[X,X^2,X^3]
# À PARTIR DES VECTEURS COLONNE X, X^2, X^3
> heures3=cbind(heures,heures^2,heures^3)
> heures3
      heures
 [1,]      0   0     0
 [2,]      2   4     8
 [3,]      4  16    64
 [4,]      6  36   216
 [5,]      8  64   512
 [6,]     10 100  1000
 [7,]     12 144  1728
 [8,]     14 196  2744
 [9,]     16 256  4096
[10,]     18 324  5832
[11,]     20 400  8000
[12,]     22 484 10648
[13,]     24 576 13824
[14,]     26 676 17576
[15,]     28 784 21952
```

```
# AJUSTEMENT LINÉAIRE MULTIPLE.
> lsfit(heures3,masse)
$coefficients
   Intercept       heures
 0.420183007  0.168389565 -0.012767745
0.001199121

# calcul de la SCE résiduelle
> sum(lsfit(heures3,masse)$residuals^2)
[1] 0.04104726
```

T.P.N 4 (SORTIES R BRUTES) : ÉTUDE DE LA CAPACITÉ D'ACCUEIL QUE DOIT AVOIR UN CENTRE DE LOISIR, EN FONCTION DE LA POPULATION DES COMMUNES CONCERNÉES

(ACTIVITÉS PLURIDISCIPLINAIRES, ANNEXE 2 DE L'ARTICLE DE S. MANGANELLI)

```
>popac=read.table("RegNL.txt",header=T,nrows=
8)
> popac
  population accueil
1       9780     102
2      20130     123
3      29670     168
4      40210     259
5      49890     354
6      61040     480
7      70120     679
8      79870     997

>nlspopac=nls(accueil~exp(a*population+b),sta
rt=list(a=.00003,b=4))
> print(nlspopac)
Nonlinear regression model
  model:  accueil ~ exp(a * population + b)
   data:  parent.frame()
          a           b
3.503911e-05 4.089063e+00
 residual sum-of-squares:  1972.852

> summary(nlspopac)
Formula: accueil ~ exp(a * population + b)
Parameters:
   Estimate Std. Error t value Pr(>|t|)
a 3.504e-05  1.010e-06   34.71 3.81e-08 ***
b 4.089e+00  7.228e-02   56.58 2.05e-09 ***
Signif. codes:  0 `***' 0.001 `**' 0.01 `*'
0.05 `.' 0.1 ` ' 1
Residual standard error: 18.13 on 6 degrees
of freedom
Correlation of Parameter Estimates:
        a
b -0.9836

> names(summary(nlspopac))
[1] "formula"      "residuals"    "sigma"
"df"           "cov.unscaled"
[6] "correlation"  "parameters"
> names(nlspopac)
[1] "m"           "data"        "call"
"dataClasses"

> (summary(nlspopac))$residuals
[1]  17.9219810   2.1681462  -0.7929878
14.8005653  11.1942979 -26.6649231
[7] -17.4562604  16.9179395
> (summary(nlspopac))$parameters

      Estimate   Std. Error  t value
Pr(>|t|)
a 3.503911e-05 1.009583e-06 34.70653
3.812164e-08
b 4.089063e+00 7.227596e-02 56.57570
2.048280e-09
> (summary(nlspopac))$formula
accueil ~ exp(a * population + b)
> (summary(nlspopac))$cov.unscaled
              a             b
a  3.099848e-15 -2.182685e-10
b -2.182685e-10  1.588709e-05
> (summary(nlspopac))$sigma
[1] 18.13308
```

```
> nlspopac$call
> nls(formula = accueil ~ exp(a * population + b),
start = list(a = 3e-05,
    b = 4), control = structure(list(maxiter = 50,
tol = 1e-05,
    minFactor = 0.0009765625), .Names = c("maxiter",
"tol", "minFactor"
)), trace = FALSE)

> nlspopac$dataClasses
   accueil population
 "numeric"  "numeric"
> sum(summary(nlspopac)$residuals^2)
[1] 1972.852
save.image("D:/HubW/MATH/FiR/popac1.RData")


> load("D:/Math/FiR/popac1.RData")
[1] ".Traceback" "nlspopac"   "popac"
> names(popac)
[1] "population" "accueil"
> par(mfrow=c(2,2),ask=T)
> attach(popac)

> plot(population,accueil,pch=11,bg="yellow")
Hit <Return> to see next plot:
> plot(population,accueil,pch=21,bg="yellow")

> predict(nlspopac)
[1]  84.07802 120.83185 168.79299 244.19943
342.80570
[6] 506.66492 696.45626 980.08206
> lines(population,predict(nlspopac))

>plot(summary(nlspopac)$residuals,type="h",col="red"
)

> abline(h=0)
>plot(population,summary(nlspopac)$residuals,type="h
",col="red")
> abline(h=0)
> help("abline")
> detach()

> #            TRANSFORMATION
> load("D:/HubW/MATH/FiR/popac1.RData")
[1] ".Traceback" "nlspopac"   "popac"
> attach(popac)
> trac=lsfit(population,log(accueil))
> trac
$coefficients
   Intercept            X
4.205839e+00 3.310053e-05
$residuals
[1]  0.095410809 -0.059968185 -0.063967664
[4]  0.020016781  0.012072454 -0.052509319
[7] -0.006227152  0.055172275
$intercept
[1] TRUE
$qr
$qt
[1] -16.117227743   2.156616138  -0.082386949
[4]  -0.009463758  -0.027566807  -0.103850002
[7]  -0.067096885  -0.015929643
```

T.P. N 4 (SUITE)

```
$qr
      Intercept             X
[1,] -2.8284271 -1.275302e+05
[2,]  0.3535534  6.515351e+04
[3,]  0.3535534  9.509798e-02
[4,]  0.3535534 -6.667380e-02
[5,]  0.3535534 -2.152460e-01
[6,]  0.3535534 -3.863803e-01
[7,]  0.3535534 -5.257434e-01
[8,]  0.3535534 -6.753900e-01
$qraux
[1] 1.353553 1.241521
$rank
[1] 2

$pivot
[1] 1 2
$tol
[1] 1e-07
attr(,"class")
[1] "qr"

> names(trac)
[1] "coefficients" "residuals"
[3] "intercept"    "qr"
> summary(trac)
             Length Class  Mode
coefficients 2      -none- numeric
residuals    8      -none- numeric
intercept    1      -none- logical
qr           6      qr     list

> trpred=log(accueil)-trac$residuals
> pred=exp(trpred)
> trpred
[1] 4.529562 4.872153 5.187932 5.536811
[5] 5.857224 6.226295 6.526848 6.849578
> pred
[1]  92.71794 130.60174 179.09773 253.86720
[5] 349.75204 505.87794 683.24143 943.48314
> sum((accueil-pred)^2)
[1] 3863.206
> plot(population,accueil,pch=21,bg="yellow")
Hit <Return> to see next plot:
> lines(population,pred,col="blue")

>
plot(population,trac$residuals,type="h",col="
red")
> abline(h=0)
> plot(population,accueil-pred,
type="h",col="red");abline(h=0)
> detach()
```

RÉFÉRENCES SUCCINCTES:

(1) JOLIVET E. (1983). Introduction aux modèles mathématiques en biologie. I.N.R.A. Actualités scientifiques et agronomiques. Masson éd.

(2) LEBRETON J.D., MILLIER C. (sous la direction de, 1982). Par E. Jolivet, C. Millier, J.D. Lebreton, A. Pavé, J.P. Vila. Modèles dynamiques et déterministes en biologie. Masson.

(3) R. Tomassone, E. Lesquoy et C. Miller, (1983). La régression. Nouveaux regards sur une ancienne méthode statistique. Masson.

(4) A.P.M.E.P. Statistiques au Lycée, volume I : aspects théoriques, volume II : aspects pratiques. À paraître en octobre 2004 (vol.I) et dans le courant de l'année scolaire 2004–2005 (vol.II).

(5) MARQUARDT D.W. (1963). An algorithm for least square estimation of non linear parameters. S.I.A.M. J., 11, 431-441.

(6) TOMASSONE R., ROUX C., (1973). Ajustements non-linéaires (HAUSS59). Note interne du Laboratoire de Biométrie du C.N.R.Z.

(7) J. W. TUKEY (1977).Exploratory Data Analysis. Addison-Wesley. 688pp. ISBN 0–201–07616–0.

(8) **R** : A Programming Environment for Data Analysis and Graphics, 1999–2004 R Development Core Team from the R-project.(www.r-project.org)

lsfit() :

(9) Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) _The New **S** Language_. Wadsworth & Brooks/Cole.

nls() :

(10) Bates, D.M. and Watts, D.G. (1988) _Nonlinear Regression Analysis and Its Applications_, Wiley

(11) Bates, D. M. and Chambers, J. M. (1992) _Nonlinear models._Chapter 10 of _Statistical Models in **S**_ eds J. M. Chambers and T. J. Hastie, Wadsworth & Brooks/Cole.