

Séance 4 : Régression Non Linéaire avec R

Exercice 1 Définition d'une fonction sous R

1. Définir la fonction f_1 de trois variables (t, a, b) par

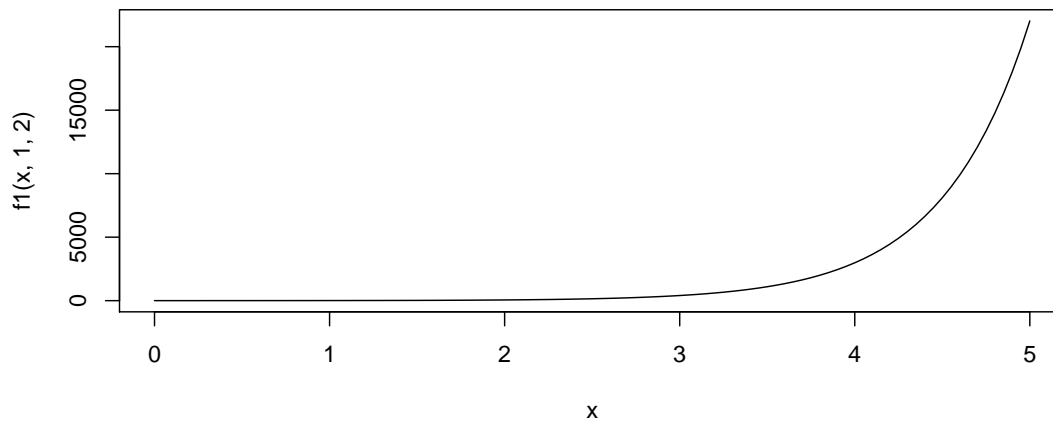
```
> f1 <- fonction(t,a,b){# paramètres d'entrée: t,a,b
+ out=a*exp(b*t)
+ return(out)}#renvoie la valeur de out
```

et la tester en calculant $f_1(0,1,2)$ puis tracer le graphe de $f_1(t,1,2)$ pour t allant de 0 à 5

```
> f1( 0,1,2)# calcul de f(0,1,2)
```

```
[1] 1
```

```
> curve(f1(x,1,2),from=0,to=5)# curve: attention la variable est toujours x
```



2. Si (a, b) sont des paramètres, on peut aussi les regrouper dans un vecteur indexé par des entiers et définir f comme fonction de (t, par) . Définir f comme suit

```
> f <- fonction(t,par){# paramètres d'entrée: t (un réel) et par (vecteur)
+ out=par[1]*exp(par[2]*t) # par contient deux valeurs
+ return(out)}
```

```
> par=c(1,2)# initialisation du vecteur par
```

```
> f(0,par)
```

```
[1] 1
```

Remarque: si (a, b) sont des paramètres, on peut aussi les regrouper dans un vecteur indexé par des noms et définir f comme fonction de (t, par) (ne pas le faire).

```
> f3 <- fonction(t,par){
+ out=par["a"]*exp(par["b"]*t)
+ return(out)}
```

```
> par=c(a=1,b=2)# initialisation du vecteur par
```

Exercice 2 Rendement d'un procédé batch dans un fermenteur et modélisation de l'évolution de la biomasse et du substrat
Après différentes calibrations, on a obtenu les concentrations contenues dans le fichier chemostat.csv (à télécharger).

On veut modéliser l'évolution des concentrations x et s sous les conditions "batch" par les équations différentielles suivantes :

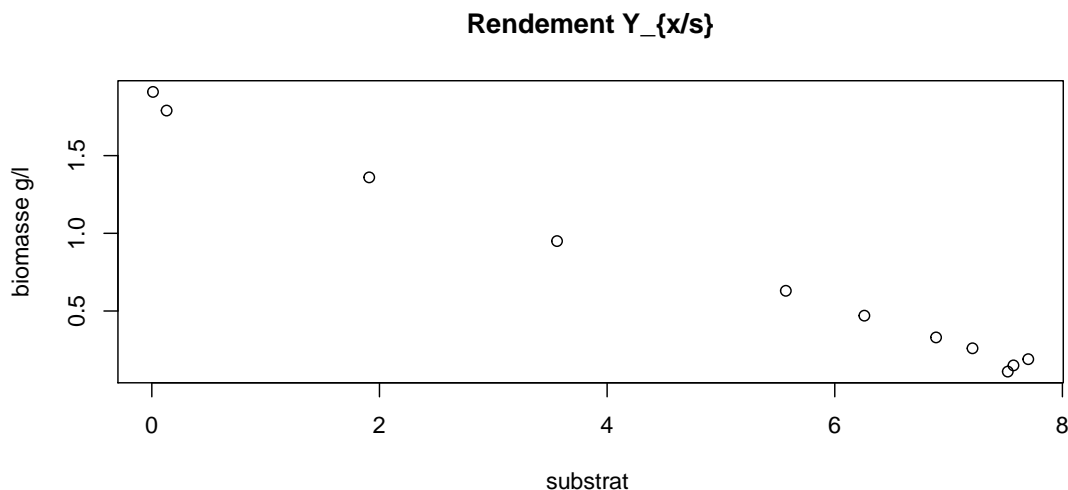
$$\begin{cases} \frac{dx}{dt} = \mu x & x(0) = x_0 \\ \frac{ds}{dt} = -\frac{\mu}{Y_{x/s}} x & s(0) = s_0 \end{cases}$$

où μ le taux de croissance de la biomasse, et $Y_{x/s}$ le rendement instantané sont supposés constants au cours du temps et (x_0, s_0) sont les concentrations initiales (des constantes strictement positives).

1. Détermination du rendement $Y_{x/s}$

(a) Lire le fichier de données chemostat.csv dans la data frame donnees. Tracer le nuage de points expérimentaux (s_i, x_i) .

```
> donnees <- read.csv2("chemostat.csv")
> attach(donnees)# pour simplifier les noms des variables
> plot(si,xi,main="Rendement Y_{x/s}", xlab="substrat ",ylab="biomasse g/l")
```



(b) Montrer (à la main!) que pour tout t , $x(t) + Y_{x/s}s(t) = x_m$ où x_m est la constante égale à $x_m = x_0 + Y_{x/s}s_0$. On a donc $x = -Y_{x/s}s + x_m$ (une droite).

(c) Déterminer, à partir des données expérimentales, les estimations de $Y_{x/s}$ et x_m et leur écart-type. Tracer la droite de régression sur le nuage des points expérimentaux. Afficher son équation sur le graphe.

```
> rend=lm(xi~si,data=donnees)
> summary(rend)

Call:
lm(formula = xi ~ si, data = donnees)

Residuals:
    Min       1Q   Median       3Q      Max
-0.09459 -0.02975  0.01868  0.02399  0.08370

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.828503   0.033175   55.12 1.07e-12 ***
si          -0.220201   0.005806  -37.93 3.06e-11 ***
---
Signif. codes:
  '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

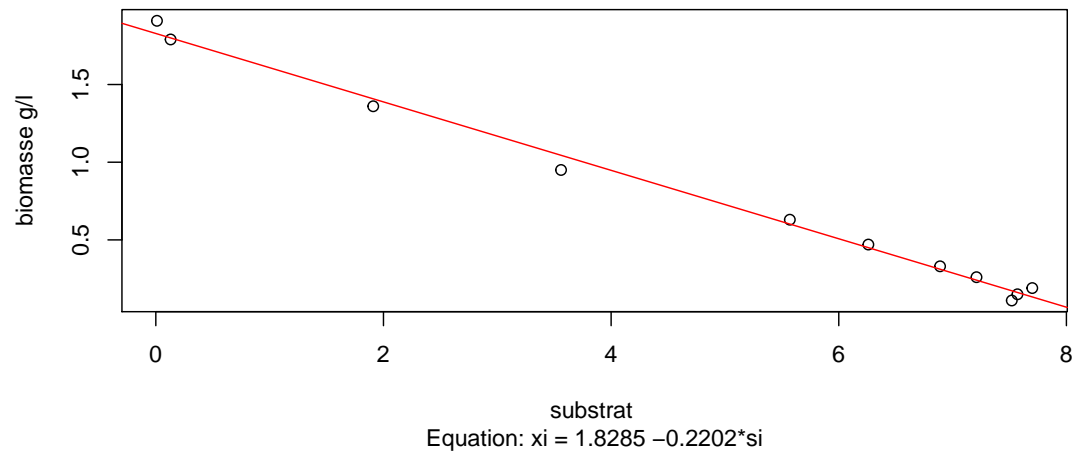
Residual standard error: 0.05533 on 9 degrees of freedom

Multiple R-squared: 0.9938, Adjusted R-squared: 0.9931

F-statistic: 1438 on 1 and 9 DF, p-value: 3.057e-11

```
> Y=-round(coef(rend)[2],4);xm=round(coef(rend)[1],4)
> eq = paste0("Equation: xi = ", round(xm,4)," -", round(Y,4),"*si" )
> plot(si,xi,main="Rendement Y_{x/s}",sub=eq, xlab="substrat ",ylab="biomasse g/l")
> abline(rend,col="red")
```

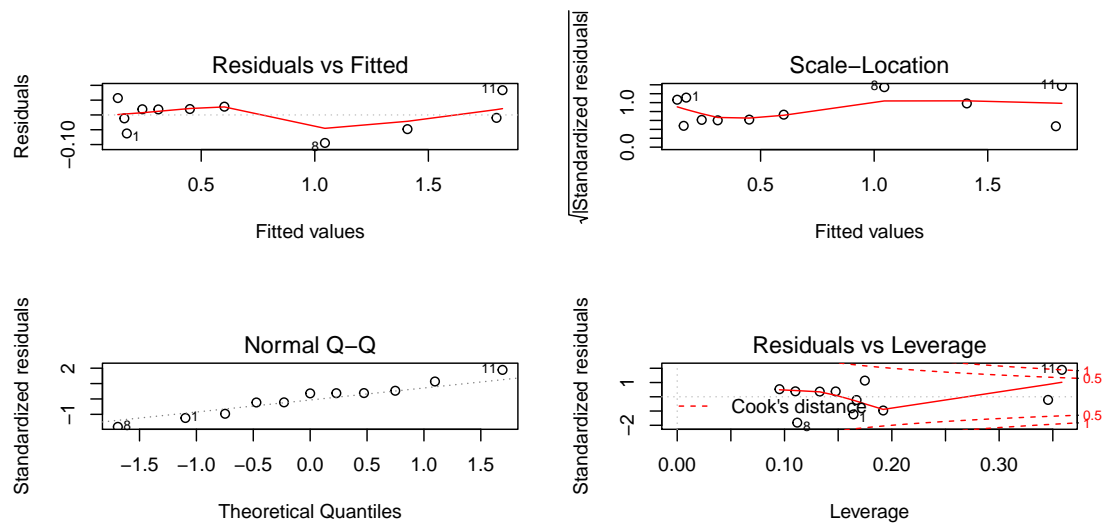
Rendement $Y_{\{x/s\}}$



Le rendement obtenu est donc environ 0.2202 et x_m calculé vaut environ 1.8285. Les coefficients de la droite sont bien déterminés (rejet de H_0).

(d) Vérifier les résidus de la régression pour valider ou non la régression.

```
> layout(matrix(1:4,2,2))# fenetre graphique coupée en 4
> plot(rend) # 4 graphiques
> layout(1)
> res <- rstudent(rend) #résidus studentisés
> plot(res,ylab="Résidus studentisés",xlab="",main="Résidus de student",ylim=c(-2.5,2.5))
> abline(h=c(-2,0,2),lty=c(2,1,2),col=c("red", "blue", "red"))
```

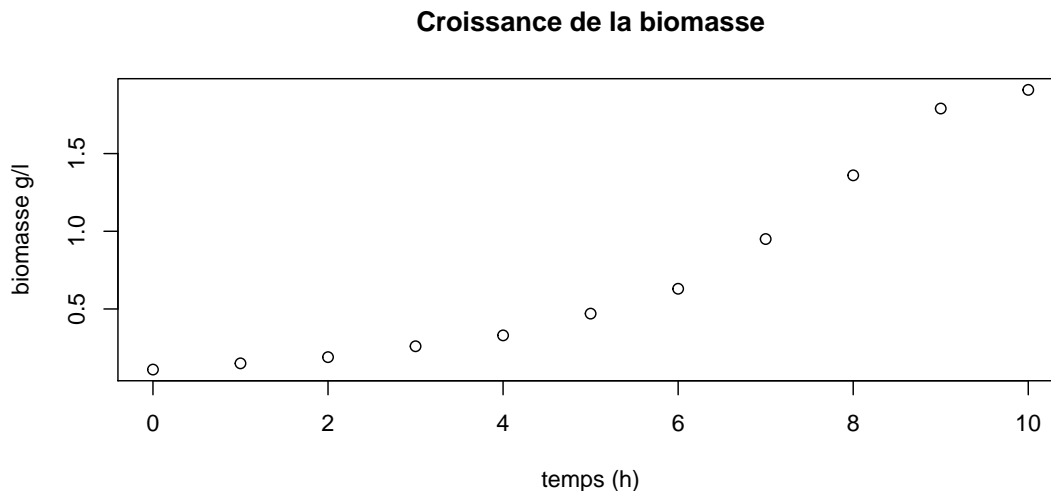


Les résidus sont acceptables: pas de véritable tendance (dupliquer les points aurait été un plus), normalité correcte. Toutefois le dernier point est suspect: distance de Cook > 1 et résidu normalisé > 2.

2. Modèle exponentiel : sélection des données correspondant au modèle

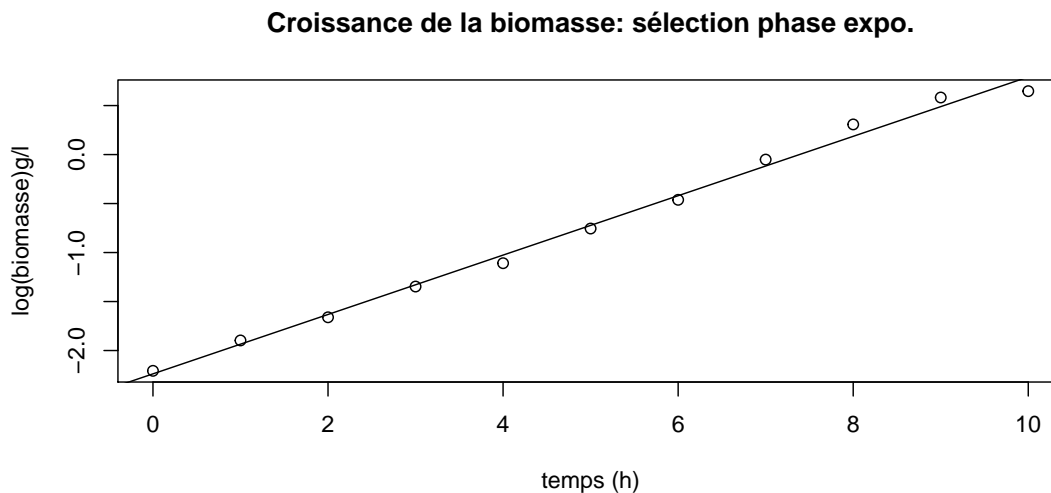
(a) Tracer le nuage de points (t_i, x_i) .

```
> plot(ti,xi,main="Croissance de la biomasse", xlab="temps (h)",ylab="biomasse g/l")
```



(b) On voit que la croissance s'infléchit en fin de graphe expérience, on peut donc supposer que la phase exponentielle se termine avant. Tracer le nuage de points $(t_i, \ln(x_i))$ pour sélectionner les points (t_i, x_i) qui "suivent" le modèle exponentiel.

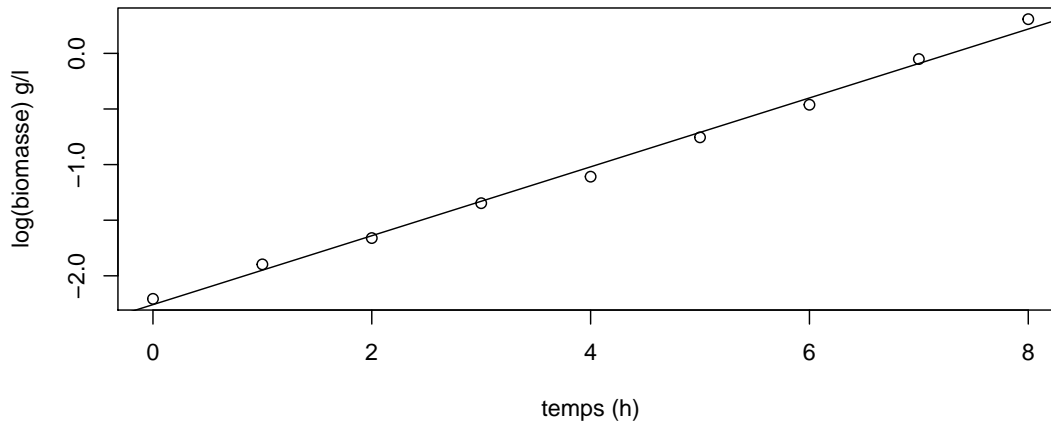
```
> plot(ti,log(xi),main="Croissance de la biomasse: sélection phase expo.",  
+      xlab="temps (h)",ylab="log(biomasse)g/l")  
> mod=lm(log(xi)~ti,data=donnees)  
> abline(mod)
```



(c) On décide de sélectionner les 8 premiers points pour déterminer les paramètres de croissance de la phase exponentielle (ce choix est subjectif). Créer une nouvelle data frame nommée *donnees2* où les deux dernières lignes de données ont été supprimées. Déterminer une initialisation pour les paramètres (μ, x_0) .

```
> donnees2=donnees[-10:-11,]  
> plot(donnees2$ti,log(donnees2$xi),main="Croissance exponentielle de la biomasse", xlab="temps (h)"  
> mod2=lm(log(xi)~ti,data=donnees2); abline(mod2)  
> a1=coef(mod2)[1];a2=coef(mod2)[2];x0=exp(a1);mu=a2
```

Croissance exponentielle de la biomasse



(d) Déterminer le taux de croissance μ à partir des données2 (t_i, x_i). On ne travaillera qu'avec les points de données2 (on élimine les points qui sortent du domaine de validité du modèle exponentiel). Vérifier graphiquement le résultat.

```
> # appel de nls: construire la suite (a_n,b_n)
> # start contient valeurs initiales des parametres (a_0,b_0)=(x0,mu) déterminés avant
> # formule xi~f(ti,c(a,b)) à chaque iteration a et b seront remplacés par (a_n,b_n)
> initialisation=list(a=x0,b=mu)
> modele <- nls(xi~ f(ti,c(a,b)), data = donnees2, start = initialisation)
> summary(modele)
```

Formula: $x_i \sim f(t_i, c(a, b))$

Parameters:

	Estimate	Std. Error	t value	Pr(> t)
a	0.085647	0.006086	14.07	2.17e-06 ***
b	0.343939	0.009952	34.56	4.40e-09 ***

Signif. codes:

0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

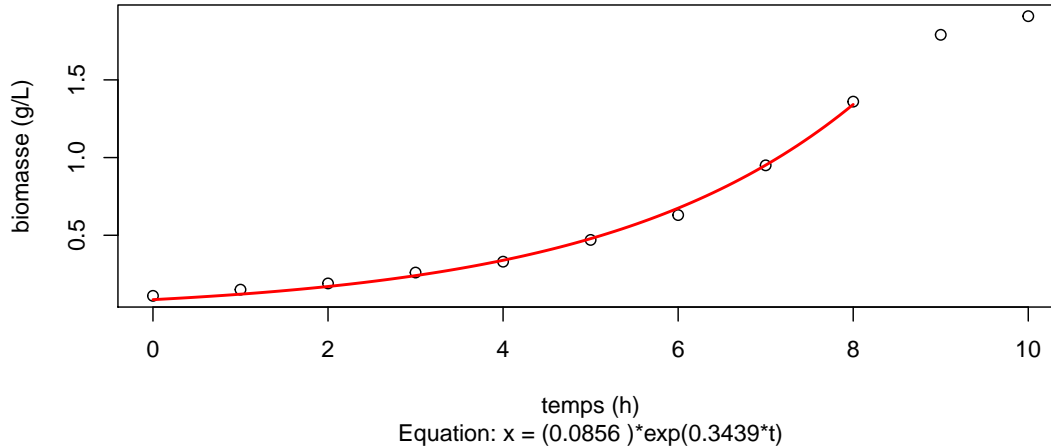
Residual standard error: 0.02583 on 7 degrees of freedom

Number of iterations to convergence: 4

Achieved convergence tolerance: 6.075e-06

```
> x0=coef(modele)[1];mu=coef(modele)[2]
> eq = paste0("Equation: x = (", round(x0,4),")*exp(", round(mu,4),"*t)")# sous titre equation
> titre=paste0("Détermination taux de croissance de la phase expo. mu=",round(mu,4))#titre du graphi
> plot(ti,xi,main=titre,sub=eq,xlab="temps (h)",ylab="biomasse (g/L)")#tracé du nuage de points
> curve(f(x,c(x0,mu)),from=0,to=8,col="red" ,lwd = 2,add=TRUE)# ajout courbe prédite
```

Détermination taux de croissance de la phase expo. $\mu=0.3439$



Il faudrait contrôler les résidus (cf séance précédente).

3. **Extension du modèle: croissance logistique (prise en compte de toutes les données)** On étudie toujours la croissance dans un chemostat en batch mais la modélisation qui suit vise à élargir le domaine de validité du modèle.

On conserve le système d'équations différentielles mais on suppose maintenant que μ le taux de croissance de la biomasse suit une fonction linéaire $\mu(s) = \lambda s(t)$ où λ est une constante positive. On suppose aussi à l'instant $t^* > 0$ le substrat est épuisé c'est-à-dire $s(t^*) = 0$ et on pose $x(t^*) = x_m$. On a toujours $x_m - x(t) = Y_{x/s}s(t)$. On conservera donc les valeurs des paramètres $Y_{x/s}$ et x_m obtenues dans la modélisation précédente.

(a) Montrer que x vérifie l'équation différentielle

$$\frac{dx}{dt} = k \left(1 - \frac{x}{x_m}\right) x$$

où $k = \frac{\lambda x_m}{Y_{x/s}}$.

(b) Montrer (à la main) que la solution de cette équation différentielle (en posant $z = \frac{1}{x}$) est

$$x(t) = \frac{x_m x_0}{x_0 + e^{-kt}(x_m - x_0)}$$

(c) Définir la fonction $g(t, \text{par}) = \frac{x_m \text{par}[1]}{\text{par}[1] + e^{-\text{par}[2]t}(x_m - \text{par}[1])}$. Déterminer avec nls les paramètres k, x_0 (x_m fixé à la valeur déterminée précédemment 1.8285).

```
> g <- fonction(t,par){# paramètres d'entrée: t (un réel) et par (vecteur)
+ out=par[1]*xm/(par[1]+(xm-par[1])*exp(-par[2]*t)) # par contient 2 valeurs
+ return(out)}
> init=list(a=0.1,c=1)
> logistique<- nls(xi ~ g(ti,c(a,c)), data = donnees, start = init)
> summary(logistique)
```

Formula: $xi \sim g(ti, c(a, c))$

Parameters:

	Estimate	Std. Error	t value	Pr(> t)
a	0.01386	0.01030	1.346	0.211
c	0.75072	0.11294	6.647	9.41e-05 ***

Signif. codes:

0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

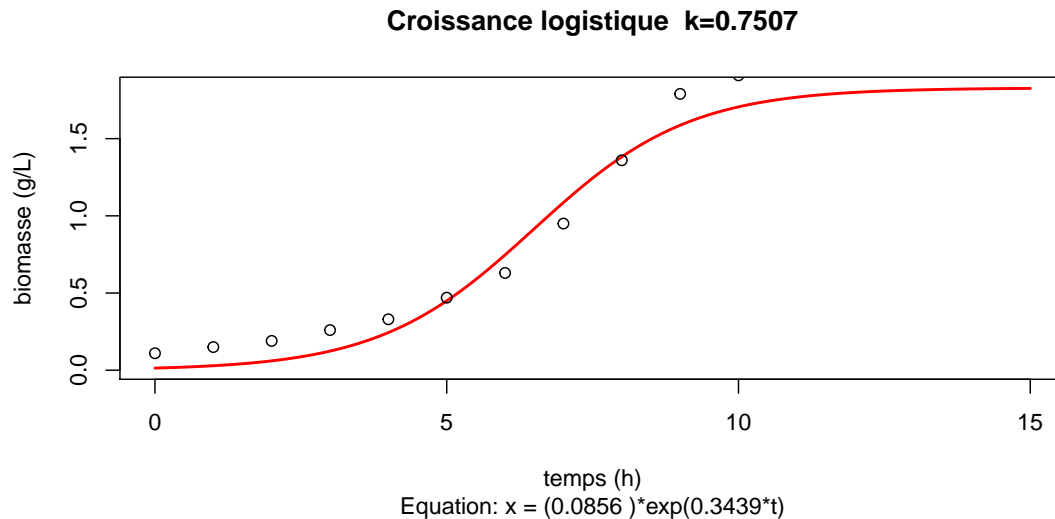
Residual standard error: 0.1425 on 9 degrees of freedom

Number of iterations to convergence: 13

Achieved convergence tolerance: 5.878e-06

(d) Tracer le nuage de points et la courbe théorique obtenue.

```
> x0=coef(logistique)[1];k=coef(logistique)[2]
> titre=paste0("Croissance logistique k=",round(k,4))#titre du graphique
> curve(g(x,c(x0,k)),from=0,to=15,col="red" ,lwd = 2,main=titre,sub=eq,
+       xlab="temps (h)",ylab="biomasse (g/L)")# courbe prédite
> points(ti,xi)#tracé du nuage de points
```



(Remarque: Il faudrait contrôler les résidus).

(e) Améliorations possibles: ajouter x_m comme paramètre ? Peut-être faut-il ajouter des bornes sur les paramètres ? (cf Help: nls)

```
> h <- fonction(t,par){# paramètres d'entrée: t (un réel) et par (vecteur)
+   out=par[1]*par[3]/(par[1]+(par[3]-par[1])*exp(-par[2]*t)) # par contient 3 valeurs
+   return(out)}
> low=c(x0,k,xm)*0.8 #bornes inf pour parametres
> up=c(x0,k,xm)*1.2 #bornes sup
> # appel nls avec algortihm=port pour pouvoir utiliser des bornes
> logistique3<- nls(xi~ h(ti,c(a,b,c)), data = donnees, start = list(a=x0,b=k,c=xm),
+                  algorithm = "port",lower=low,upper=up)
> summary(logistique3)
```

Formula: $x_i \sim h(t_i, c(a, b, c))$

Parameters:

	Estimate	Std. Error	t value	Pr(> t)
a	0.01663	0.01162	1.432	0.190064
b	0.67845	0.11834	5.733	0.000437 ***
c	2.19420	0.25430	8.628	2.52e-05 ***

Signif. codes:

0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

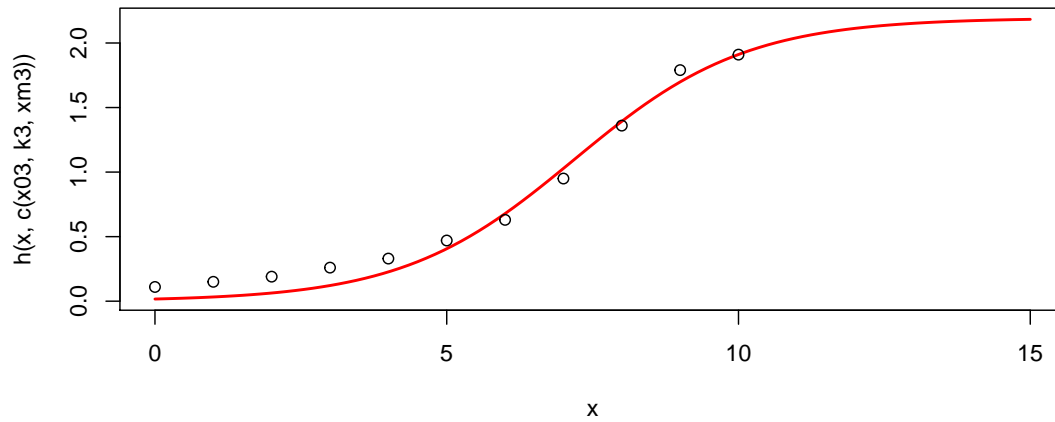
Residual standard error: 0.1064 on 8 degrees of freedom

Algorithm "port", convergence message: both X-convergence and relative convergence (5)

```

> x03=coef(logistique3)[1];k3=coef(logistique3)[2];xm3=coef(logistique3)[3]
> curve(h(x,c(x03,k3,xm3)),from=0,to=15,col="red",lwd=2)# ajout courbe prédite
> points(ti,xi,main=titre,sub=eq,xlab="temps (h)",ylab="biomasse (g/L)")#tracé du nuage de points

```



Exercice 3 On introduit maintenant des bactéries dans le fermenteur. La concentration en oxygène dissous dans un fermenteur en régime transitoire est donnée par l'équation

$$\frac{dO_{2L}}{dt} = KLa(O_{2L}^* - O_{2L}) - Q_{O_2}X \quad O_{2L}(0) = O_{2L_0}$$

où O_{2L}^* est la concentration saturante d'oxygène liquide à 30°C , X la concentration bactérienne du milieu, et Q_{O_2} la vitesse spécifique de respiration des bactéries. On suppose que $O_{2L}^* = 100$. A l'aide de données expérimentales ci-dessous, on veut déterminer KLa , $Q_{O_2}X$ et O_{2L_0} .

t_i temps (s)	10	20	30	40	50	60	70	100	130
O_{2L_i} (mg l^{-1})	43,5	53,5	60	67,5	70,5	72	73	73,5	73,5

1. Tracer le nuage de points expérimentaux (t_i, O_{2L_i}) .
2. Résoudre l'équation différentielle vérifiée par O_{2L} . On montrera que $O_{2L}(t) = (O_{2L}(0) - \overline{O_{2L}})e^{-KLat} + \overline{O_{2L}}$ où $\overline{O_{2L}} = O_{2L}^* - \frac{Q_{O_2}X}{KLa}$.
3. Déterminer les paramètres.
4. Tracer la courbe théorique obtenue et tracer les résidus dans un autre graphique.