

# Apresentação do R com um exemplo de análise de regressão não-linear

Eduardo Esteves<sup>1</sup>

<sup>1</sup>Universidade do Algarve

28 de Outubro de 2020

## Abstract

No seguimento dum artigo anterior acerca da análise de regressão não-linear (simples) utilizando a ferramenta Solver<sup>®</sup> do Excel<sup>®</sup> propõe-se neste artigo a apresentação do R, uma linguagem de programação e um ambiente para computação estatística e gráfica, através da sua aplicação na “descrição” de relações estatísticas (não-lineares) entre variáveis.

**Palavras-chave:** R, Regressão não-linear.

## Introdução

O R ([R Core Team and R Development Core Team, 2017](#)) é ao mesmo tempo uma linguagem de programação e um ambiente para computação estatística e gráfica. Trata-se de uma linguagem de programação especializada em computação com dados. Uma das suas principais características é o seu carácter gratuito e a sua disponibilidade para uma gama bastante variada de sistemas operativos (vd. ). Apesar do seu carácter gratuito, o R é uma ferramenta bastante poderosa com boas capacidades ao nível da programação e um conjunto bastante vasto (e em constante crescimento) de packages que acrescentam bastantes potencialidades à já poderosa versão base do R ([Torgo, 2006](#)). Contudo, o termo “ambiente” pretende caracterizar o R como um sistema completo e coerente ao invés dum conjunto de ferramentas muito específicas e relativamente inflexíveis ([Venables, Smith and Team, 2006](#)).

O R pode ser entendido como uma implementação da linguagem S desenvolvida por Rick Becker, John Chambers e Allan Wilks nos Bell Laboratories (E.U.A.), que também constitui a base do software S-Plus<sup>®</sup> (Insightful Corp.). A evolução da linguagem S está descrita em quatro livros de John Chambers e colaboradores. As “distribuições” do R incluem um bom conjunto de manuais (vd. ) e existem, actualmente, vários livros que descrevem a utilização do R para análise estatística de dados (vd. ), *e.g.* ([Dalgaard, 2002](#)) ou ([Crawley, 2007](#)). Na última revisão deste artigo usei a versão 2.12.0 (embora sejam disponibilizadas regularmente “novas” versões do R). O ficheiro de instalação (R-X.XX.X-win32.exe) para Microsoft Windows<sup>®</sup> pode ser obtido em e o processo de instalação é simples. Também existem “versões” para Mac OS e Unix/Linux, e 64-bit para Windows. O sítio electrónico do projecto constitui a principal referência do R e funciona como ponto de partida para explorar mais este sistema.

Boa parte das pessoas utiliza o R como um sistema para análise estatística de dados, uma vez que a maioria das “estatísticas clássicas” e muitas das metodologias mais recentes estão disponíveis, embora os promotores prefiram “pensar” o R como um ambiente no qual essas técnicas têm sido implementadas ([Venables, Smith and Team, 2006](#)). Cerca de 25 pacotes (*i.e.* conjuntos de funções) fazem parte do sistema básico (os “re-

commended” packages), mas muitos outros estão disponíveis através do CRAN (via para instalação. Uma importante diferença entre a linguagem S (e o R) e outros sistemas (e.g. SPSS® ou SAS®) reside na (muito) menor quantidade de resultados apresentados para uma qualquer análise estatística, embora os resultados sejam guardados em objectos para posterior consulta ou utilização pelo R noutras funções (Venables, Smith and Team, 2006).

Neste artigo, pretende-se apresentar (muito sucintamente) o R, uma linguagem de programação especializada em computação com dados, utilizando-a para analisar problemas cujo objectivo é “descrever” relações estatísticas (não-lineares) entre variáveis. Não se pretende discutir aqui os aspectos estatísticos dos resultados obtidos, apenas o *modus operandi* (ainda que de forma simplificada).

## Regressão não-linear (simples)

Os vários aspectos relacionados com a regressão não-linear (simples) que servem de base a este artigo foram abordados, ainda que de forma informal e possivelmente incompleta, num artigo anterior (Esteves, 2011). Em vários domínios do conhecimento, e.g. biologia, física, química, engenharia, etc., são usados modelos matemáticos para descrever um conjunto de dados empíricos, genericamente

$$y = f(\theta) + \epsilon$$

em que  $y$  é a variável dependente,  $x$  é a variável “independente” – por vezes, controlada pelo investigador – e  $f(x)$  é uma função que pode incluir um ou mais parâmetros  $\theta$ , e  $\epsilon$  são os erros aleatórios, independentes e com distribuição normal. Outra formulação, equivalente, é  $\hat{y} = f(x)$  (em que  $\hat{y}$  se lê valor esperado, ou estimado, de  $y$ ). Quanto melhor  $f(x)$  se ajustar aos dados, mais “rigorosamente” descreverá aquela relação (Brown, 2001). Pretende-se ajustar a função  $f(x)$  aos dados empíricos de forma a minimizar os erros  $\epsilon_i = (y_i - \hat{y}_i)$ . De facto, o objectivo é estimar o(s) parâmetro(s) da função  $f(x)$  de modo a minimizar a soma dos quadrados dos erros, SQE – método dos mínimos quadrados. No caso de funções (ou modelos) não-lineares, e.g.  $y = a \exp(b \cdot x)$ , não é possível obter as estimativas dos parâmetros num único passo, como no caso de regressões lineares, pelo que a SQE é minimizada através dum processo iterativo (cíclico) utilizando um algoritmo apropriado que necessita dos valores iniciais dos parâmetros  $\theta_0$  (Bowen and Jerman, 1995). Tradicionalmente, transformam-se as variáveis de alguns modelos não-lineares de forma a linearizar a relação e a permitir a sua análise através da regressão linear. Contudo, esta abordagem é válida se a(s) variável(is) transformadas se verificam os pressupostos da análise de regressão linear.

## Exemplo

A utilização do R com um exemplo concreto (entretanto abordado por (Esteves, 2011) usando a ferramenta Solver® do Microsoft Excel®) permitirá mostrar o funcionamento e demonstrar as capacidades do software. Como se imagina, as funcionalidades não se esgotam no que aqui se apresenta.

O aspecto do R ao iniciar uma sessão de trabalho em ambiente Windows® ilustra-se na Figura . Para além da barra de ferramentas no topo (que permite realizar as tarefas comuns: abrir/gravar ficheiros, cortar/colar texto, instalar pacotes, “gerir” as janelas, etc.) surge uma janela (*R Console*) na qual se introduzem os comandos, a seguir ao sinal `>`.

Admita-se que os dados que se pretendem analisar estão num ficheiro do Excel®. Em primeiro lugar, será necessário guardar uma versão (\*.txt ou \*.csv) - a extensão txt diz respeito a ficheiros de texto separado por tabulações enquanto a extensão csv está relacionada com ficheiros de texto separados por vírgulas - desse ficheiro utilizável pelo R (devem usar-se pontos, em vez de vírgulas, como separadores decimais).

Na consola do R escrever (fazendo Enter no fim),

```
> dados<-read.table(file="G:/MyArticles/ArtigosRegressaoNaoLinearExcel/RegressaoNaoLinear.txt\selectlan
```

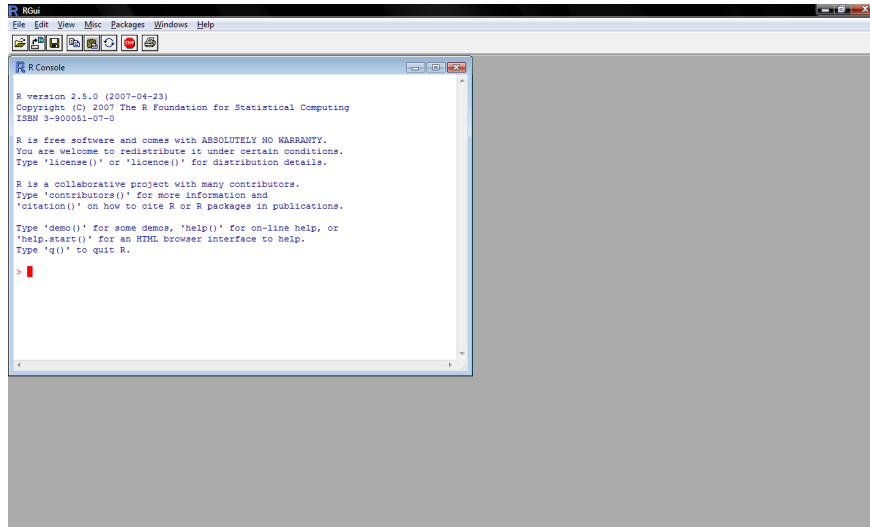


Figura 1: Aspeto da janela \*R Console\*.

para “carregar” a informação contida no ficheiro `RegressaoNaoLinear.txt` para o objecto `dados`. (atente-se na notação `>`). Os parâmetros `header` e `sep` permitem considerar os rótulos dos dados e o tipo de ficheiro de texto (neste caso separado por tabulações), respectivamente. Por sinal, a quantidade de dados envolvida poderia muito bem ser introduzida directamente no R através de

```
> CO2<-c(141,172,181,227,309,414,641, 936)
> Prop<-c(0.25,0.34,0.34,0.68,0.85,0.99,0.98,0.99)
```

em que `c(...)` permite concatenar um conjunto de dados num objecto (neste caso, `CO2` e `Prop`). Podemos visualizar a informação usando a função `plot`

```
> plot(Prop~CO2, las=1,xlim=c(0,1000),ylim=c(0,1),pch=16,cex=1.5, xlab=\selectlanguage{english}"CO2 (ppm)"))
```

em que `Prop~CO2` descreve o “modelo” que se pretende representar (neste caso `Prop` em funcao de `CO2`) e cujos restantes parametros especificam as varias caracteristicas do grafico (`las` para orientacao dos rotulos do eixo dos `yy`, `xlim` e `ylim` para determinar os limites dos eixos, `pch` para definir o simbolo usado, `cex` para aumentar o tamanho relativo dos caracteres, e `xlab` e `ylab` para “legendar” os eixos) (Figura a). Para mais informacoes acerca desta (ou qualquer outra) funcao pode fazer-se

```
> plot
```

(ou entao procurar com `>help.search(“termo”)` ou, ainda, usar a opcao `Help> Apropos...`). Usando o botao direito do rato sobre o grafico, e possivel copiar a figura (como meta-ficheiro `Windows(r)` por exemplo) para inclusao num ficheiro do `Microsoft Word(r)`. Admita-se (por razoes scientificas) que um modelo “dose-resposta” (logistico modificado)  $P_{\{i\}} = \frac{1}{3} + \frac{\exp(B(T-\log(x)))}{(1+\exp(T-\log(x)))}$  em que  $P_i$  e a proporcao de respostas positivas do provador  $i$ ,  $\log(x)$  e o logaritmo do valor do estimulo  $x$ ,  $B$  e o “declive” e  $T$  e o limiar para o provador  $i$  (em  $\log(x)$ ), e adequado para descrever os dados. Para ajustar o modelo basta fazer

```
> ajuste<-nls(Prop~1/3+exp(B*(T-log10(CO2)))/(1+exp(B*(T-log10(CO2))))),data=data.frame(CO2=CO2,Prop=Prop))
```

em que o objecto `ajuste` inclua os resultados duma regressao `> nao-linear` (atraves da funcao `nls`) - outras funcoes (muito) usadas sao `lm` e `aov` para regressao linear (i.e. modelos lineares) e analise de variancia, respectivamente - do modelo - a definicao dos modelos em S esta minuciosamente descrita em (Chambers, Hastie and Pregibon, 1990) `Prop~(1/3+exp(B*(T-log10(CO2)))/(1+exp(B*(T-log10(CO2))))` considerando os dados (indicados para o parametro `data` sob a forma de uma `data.frame` - neste caso, considerou-se que

o utilizador introduziu directamente no sistema os dados ao inves de “carregar” o ficheiro dos dados (sendo que bastaria apenas indicar data=dados, no comando) - e as estimativas iniciais dos parametros do modelo descritas em start (atraves duma list). Os resultados do ajuste (Figura 2) podem ser consultados fazendo

```
> summary(ajuste)
```

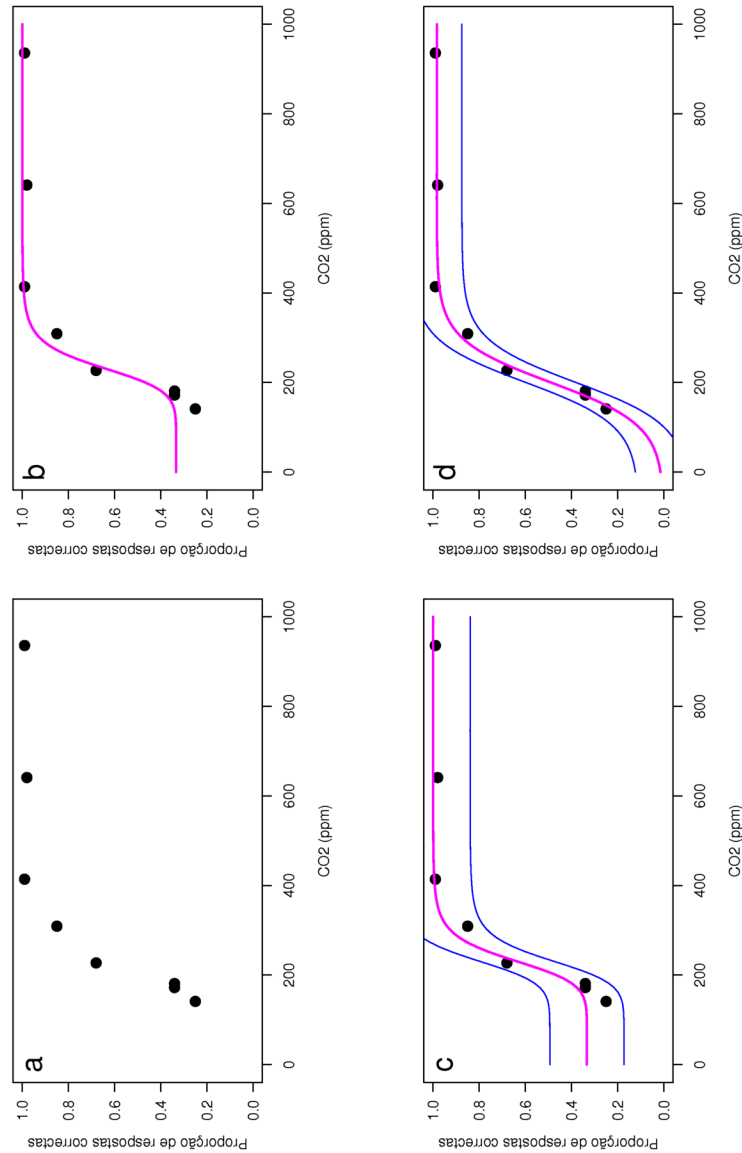


Figura 2: Diagramas de dispersão da proporção de respostas vs. concentração de CO2: (a) dados originais; (b) modelo “dose-resposta” (logístico modificado); (c) modelo “dose-resposta” com intervalos de 95% de confiança; (d) e modelo logístico (ver texto principal).

É possível ilustrar o ajuste, obtendo primeiro as estimativas da variável Prop (através da função predict) e “desenhando-as”, em seguida, sobre o gráfico entretanto obtido (Figura 2) ou seja

```
> PropEsp<-predict(ajuste,newdata=list(CO2=seq(0,1000,length=100)),se.fit=T)
> lines(PropEsp~seq(0,1000,length=100),lwd=2,col=6)
```

A informação contida no objecto ajuste permite obter as estimativas para 100 valores de CO2 entre 0 e 1000 (daí o parâmetro seq(1,1000,length=100) no comando acima) assim como os respectivos erros-padrão (se.fit=T) que serão guardadas no objecto designado PropEsp. A adição duma linha espessa e magenta (lwd=2 e col=6) à Figura a é relativamente fácil (Figura b).

```
> # Diagrama dispersão
>
> plot(Prop~CO2, las=1, xlim=c(0,1000),ylim=c(0,1),pch=16,cex=1.5,
+ xlab="CO2 (ppm)", ylab="Proporção de respostas correctas")
>
>
> ajuste<-nls(Prop~(1/3+exp(B*(T-log10(CO2))))/(1+exp(B*(T-log10(CO2)))),
+ data=data.frame(CO2=CO2,Prop=Prop),start=list(B=-10,T=2.5))
> summary(ajuste)

Formula: Prop ~ (1/3 + exp(B * (T - log10(CO2)))) / (1 + exp(B * (T - log10(CO2))))

Parameters:
      Estimate Std. Error t value Pr(>|t|)
B -19.56317     6.12626  -3.193  0.0188 *
T   2.37195     0.01908 124.324 1.83e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.06565 on 6 degrees of freedom

Number of iterations to convergence: 21
Achieved convergence tolerance: 6.796e-06

>
```

Figura 3: Aspeto da execução de comandos e respectivos resultados na janela *R Console*.

Podem, ainda, obter-se e “desenhar-se” os intervalos de confiança (upIC e loIC) que se calculam através de  $y \pm t[?]se(Y)$ , recorrendo as funcoes qt (para obter os valores “teóricos” de t-Student e cujos parametros sao  $1 - \frac{\alpha}{2}$ , neste caso 0.975, e os graus de liberdade do erro, que se obtem de summary(ajuste)\$df[2]), summary(ajuste)\$sigma para obter o erro-padrão das estimativas e lines, respectivamente (Figura

```
> upIC<-PropEsp+qt(.975,summary(ajuste)\$df[2])\*summary(ajuste)\$sigma
> loIC<-PropEsp-qt(.975,summary(ajuste)\$df[2])\*summary(ajuste)\$sigma
> lines(upIC~seq(0,1000,length=100),col=4)
> lines(loIC~seq(0,1000,length=100),col=4)
```

O modo mais simples de “testar” a bondade do ajuste e representar graficamente os dados e o modelo ajustado (bem como os respectivos intervalos de confiança), de modo a verificar (visualmente) se os parametros obtidos numericamente descrevem, de facto, a relação entre variáveis (Figura c). Por outro lado, a análise gráfica dos resíduos, e.g. resíduos vs. valores observados de X, permite “verificar” se o modelo é adequado (os resíduos representam apenas o erro experimental se não apresentam tendências ou padrões). Sendo assim, obter os resíduos através de

```
> residuos
> summary(ajuste)\$resid
```

e, em seguida, representar esse grafico em conjunto com o *normal probability plot* dos residuos (Figura )

```
> par(mfrow=c(2,1),pty=\selectlanguage{english}"s\selectlanguage{english}",las=1)
> plot(residuos\~predict(ajuste),pch=16,col=3,xlab="E{Proporcao de respostas corretas}\selectlanguage{english}"
> abline(h=0)
> qqnorm(residuos,pch=16,col=2,xlab= \selectlanguage{english}"Quantis da Distribuicao Normal\selectlanguage{english}"
> ylab=\selectlanguage{english}"Quantis Amostrais\selectlanguage{english}",main=\selectlanguage{english}"Quantis Amostrais\selectlanguage{english}"
> qqline(residuos)
> par(op)
```

>> O comando par permite estabelecer o “esquema” da figura, neste caso, > os graficos ficam “arranjados” em 2 linhas por 1 coluna – mfrow=c(2,1) > –, assim como definir a respectiva forma (quadrada, atraves de pty=“s” > - “s” de “square”). Ao grafico dos residuos *vs.* valores esperados de > Prop (atraves de plot) acrescentamos uma linha horizontal ao nivel de > zero (atraves de abline) e, por fim, desenhamos o *normal probability > plot* dos residuos (com qqnorm e qqline).

Figure 2 Diagramas de dispersão da proporção de respostas *vs.* concentração de CO2: dados originais (a); modelo “dose-resposta” (logístico modificado) (b); modelo “dose-resposta” com intervalos de 95% de confiança (c); e modelo logístico (ver texto principal) (d).

É possível ilustrar o ajuste, obtendo primeiro as estimativas da variável Prop (através da função predict) e “desenhando-as”, em seguida, sobre o gráfico entretanto obtido (Figura 2) ou seja

```
> > PropEsp<-predict(ajuste,newdata=list(CO2=seq(0,1000,length=100)),se.fit=T) > > li-
nes(PropEsp~seq(0,1000,length=100),lwd=2,col=6)
```

A informação contida no objecto ajuste permite obter as estimativas para 100 valores de CO2 entre 0 e 1000 (daí o parâmetro seq(1,1000,length=100) no comando acima) assim como os respectivos erros-padrão (se.fit=T) que serão guardadas no objecto designado PropEsp. A adição duma linha espessa e magenta (lwd=2 e col=6) à Figura a é relativamente fácil (Figura b).

Figure 3 Aspecto da execucao de comandos e respectivos resultados na janela *R Console*.

Podem, ainda, obter-se e “desenhar-se” os intervalos de confianca (upIC e loIC) que se calculam atraves de  $y \pm t_{[?]}se(Y)$ , recorrendo as funcoes qt (para obter os valores “teoricos” de t-Student e cujos parametros sao  $1[?]-[?]a[?]/[?]2$ , neste caso 0.975, e os graus de liberdade do erro, que se obtêm de summary(ajuste)\$df

]2[

), summary(ajuste)\$sigma para obter o erro-padrão das estimativas e lines, respectivamente (Figura

```
> > > upIC<-PropEsp+qt(.975,summary(ajuste)$df
```

]2[

```
)*summary(ajuste)$sigma > > > > loIC<-PropEsp-qt(.975,summary(ajuste)$df
```

]2[

```
)*summary(ajuste)$sigma > > > lines(upIC~seq(0,1000,length=100),col=4) > > > li-
nes(loIC~seq(0,1000,length=100),col=4) > > O modo mais simples de “testar” a bondade do ajuste
é representar > graficamente os dados e o modelo ajustado (bem como os respectivos > intervalos de
confiança), de modo a verificar (visualmente) se os > parâmetros obtidos numericamente descrevem, de
facto, a relação entre > variáveis (Figura c). Por outro lado, a análise gráfica dos resíduos, > e.g. resíduos
vs. valores observados de X, permite “verificar” se > o modelo é adequado (os resíduos representam apenas
o erro > experimental se não apresentam tendências ou padrões). Sendo assim, > obter os resíduos através
de > > > residuos<-summary(ajuste)$resid > > e, em seguida, representar esse gráfico em conjunto
```

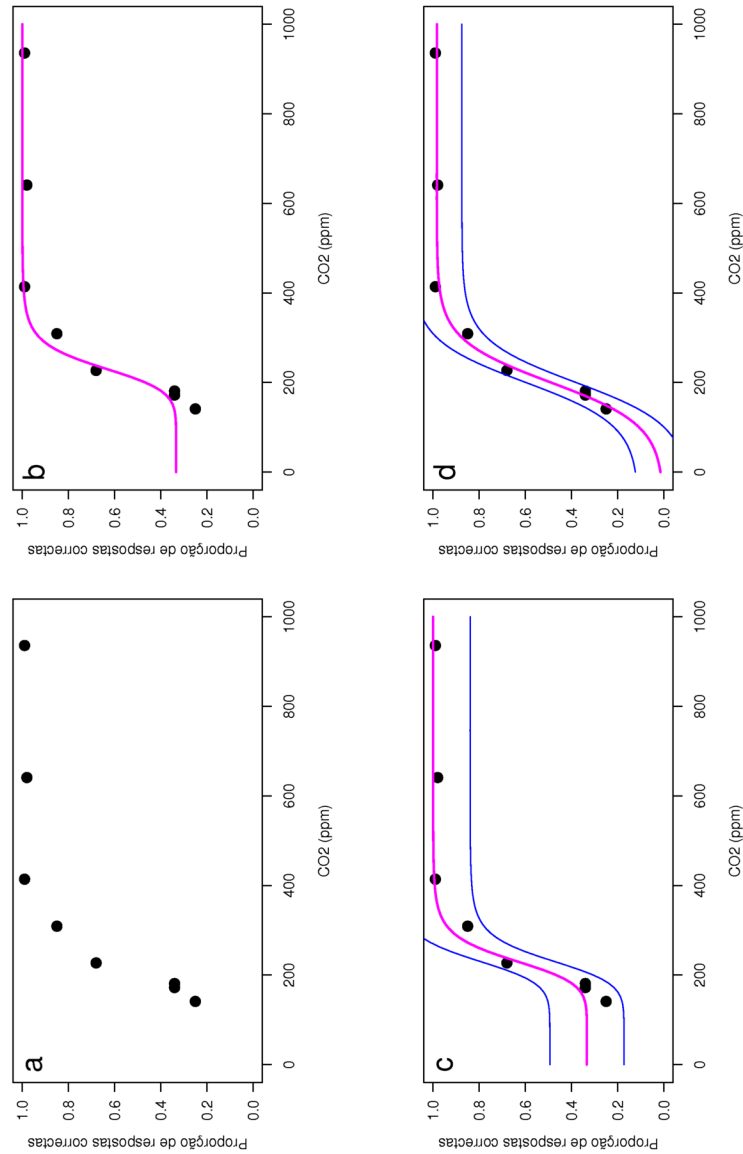


Figura 4: This is a caption

com o *normal > probability plot* dos resíduos (Figura ) > > > `op<-par(mfrow=c(2,1),pty="s",las=1)`  
 > > > `plot(resíduos~predict(ajuste),pch=16,col=3,xlab="E{Proporção de > respostas > > corretas}",ylab="Resíduos")`  
 > > > `abline(h=0)` > > > `qqnorm(resíduos,pch=16,col=2,xlab="Quantis da Distribuição > Normal", > > ylab="Quantis Amostrais",main="")` > > > `qqline(resíduos)` > > > `par(op)`  
 > > O comando `par` permite estabelecer o "esquema" da figura, neste caso, > os gráficos ficam "arranjados" em 2 linhas por 1 coluna - `mfrow=c(2,1)` > -, assim como definir a respectiva forma (quadrada, através de `pty="s"` > - "s" de "square"). Ao gráfico dos resíduos *vs.* valores esperados de > Prop (através de `plot`) acrescentámos uma linha horizontal ao nível de > zero (através de `abline`) e, por fim, desenhamos o *normal probability > plot* dos resíduos (com `qqnorm` e `qqline`).

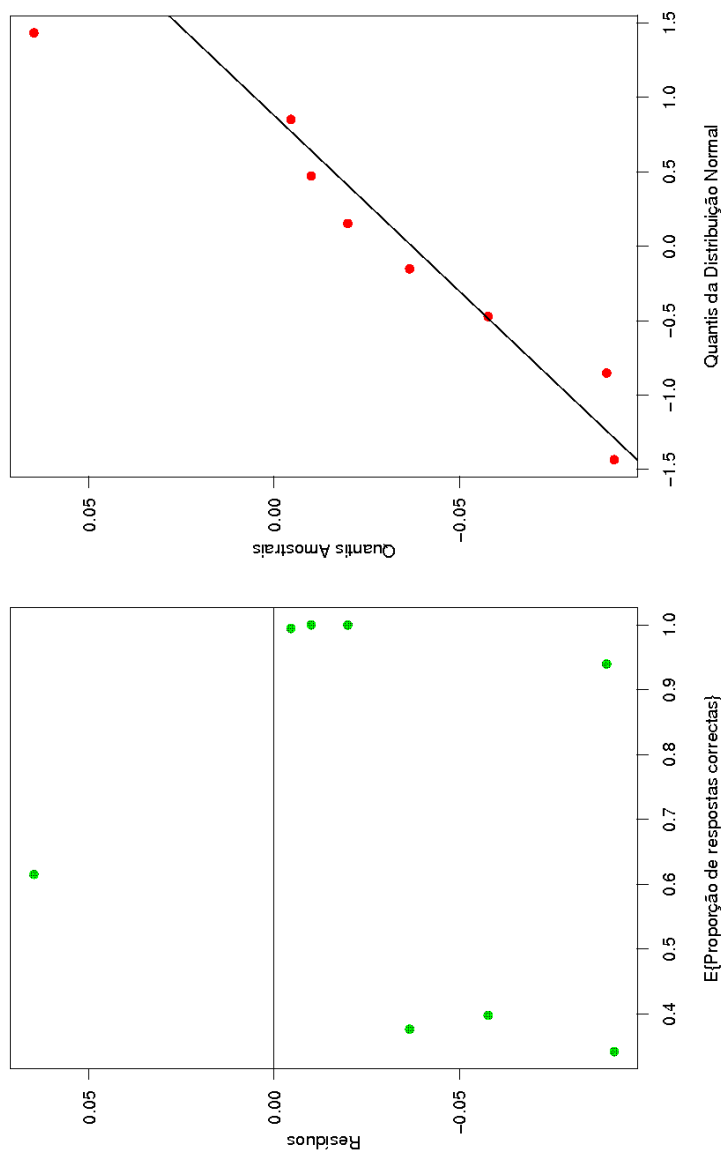


Figura 5: Análise gráfica de resíduos: resíduos *vs.* valores esperados (esq.); e *normal probability plot* dos resíduos (dir.)

É possível testar se os resíduos se distribuem normalmente através, por exemplo, de

```
> shapiro.test(residuos)
```

O resultado deste comando é

Shapiro-Wilk normality test

data: residuos



W = 0.936, p-value = 0.5719

O teste de sequências (ou “runs test” nos manuais anglófonos) permite testar, de forma simples e robusta, se os pontos (dados) diferem sistematicamente da curva ajustada (modelo) complementando a informação do diagrama dos resíduos *vs* valores observados de  $X$ .

*]alertaparaareduziapotênciaestatísticadotestesbaseados > emsequências.NoR, seránecessáriocarregaralguns pacotes.*

(Figura d)

]F[

]F[

Esta figura foi obtida através de

```
> plot(Prop~CO2,las=1,xlim=c(0,1000), ylim=c(0,1),pch=16,cex=1.5, xlab="CO2  
(ppm)",ylab="Propor
```

ção de respostas correctas“)

```
> PropEsper<-predict(ajustes,newdata= list(CO2=seq(0,1000,length=100)),se.fit=T)
```

```
> lines(PropEsper~seq(0,1000,length= 100),lwd=2,col=6)
```

```
> uppIC<-PropEsper+qt(.975,summary(ajustes)$df
```

]2[

```
)*summary(ajustes)$sigm
```

```
> lowIC<-PropEsper-qt(.975,summary(ajustes)$df
```

]2[

```
)*summary(ajustes)$sigm
```

```
> lines(uppIC~seq(0,1000,length=100), col=4)
```

```
> lines(lowIC~seq(0,1000,length=100), col=4)
```

, de formulação mais generalizada do que o modelo “dose-resposta” anterior, através de

```
> ajustes<-nls(Prop~C/(1+A*exp(-B*CO2)),data=data.frame(CO2=CO2,Prop=Prop),  
start=list(A=10,B=.0025,C=1))
```

```
> summary(ajustes)
```

os resultados do teste de F obtêm-se facilmente através de

```
> anova(ajuste,ajustes)
```

Os passos descritos anteriormente, i.e. uma sessão de trabalho (Figura ), podem (e, na minha opinião, devem) ser “guardados”. Para isso, criar um ficheiro de *script* (através de File>New script), de facto um ficheiro texto, onde são “guardados” os comandos. Este ficheiro pode ser carregado noutra sessão de trabalho (através de File>Open script...) e as análises e gráficos podem ser repetidos n-vezes (janela *R Editor*). O *script* do exemplo descrito aqui pode ser obtido no sítio electrónico do autor.

Para terminar uma sessão de trabalho, primeiro “limpar” da memória os objectos criados (com a função `rm`) e, depois, encerrar o programa (geralmente não guardo uma “imagem” do ambiente de trabalho):

```
> rm(list=ls()) ; q()
```

Figure 5 Aspecto dum sessão de trabalho no R.

#### Considerações finais

As “dificuldades estatísticas” relacionadas com a utilização do Excel(r) da Microsoft (e da respectiva ferramenta Solver(r)) para análise de regressão não-linear

*]limitamasuautilizacaoaoscasos” mais > simples” ou para feitos pedagogicos. Em situações” profissionais” > deveuse*

Embora as características do R tornem (mais) difícil a aprendizagem, o seu carácter gratuito, o facto de ser uma ferramenta bastante poderosa com boas capacidades ao nível da programação e um conjunto bastante vasto (e em constante crescimento) de \*packages\* que acrescentam bastantes potencialidades estatísticas e gráficas, o crescente interesse de utilizadores das mais variadas formações (que se reúnem anualmente desde 2004 nas conferências useR! – em 2011 a useR!2011 decorreu no Reino Unido, \*cf\*. <http://www.warwick.ac.uk/statsdept/useR-2011/>) e a disponibilidade dos promotores (R Development Core Team) e da “comunidade” de utilizadores, mais ou menos avançados, ajudam (e muito) esse processo. Os vários manuais (<http://cran.r-project.org/manuals.html>), a página wiki (<http://wiki.r-project.org/>) e a \*mailing list\* de ajuda (<http://stat.ethz.ch/mailman/listinfo/r-help>) permitem ultrapassar a (esmagadora) maioria das dificuldades surgidas durante a utilização do R.

∩ Agradecimentos ∩ ∩ Este artigo muito beneficiou dos comentários, correções e sugestões ∩ dos colegas e leitores. ∩ ∩ Referências ∩ ∩ [

]1[

A. M. Brown: “A step-by-step guide to non-linear regression analysis of experimental data using a Microsoft Excel spreadsheet”, *Computer methods and Programs in Biomedicine*, pp. 191-200, 2001.

]2[

John M. Chambers, Trevor J. Hastie: *Statistical models in S*. Chapman & Hall, 1992.

]3[

W.J. Conover: *Practical nonparametric statistics*. John Wiley & Sons. Inc., 1999.

]4[

M.J. Crawley: *The R book*. Wiley, 2007. URL <http://www.google.com/books?id=8D4HVx0apZQC>.

]5[

Peter Dalgaard: *Introductory statistics with R*. Springer-Verlag New York Inc., 2002.

]6[

E. Esteves: *Regressão não-linear utilizando a ferramenta Solver(r) do Microsoft Excel(r)*. 2010. URL <http://w3.ualg.pt/~eesteves..>

]7[

R Development Core Team: *R: A Language and Environment for Statistical Computing*. 2007. URL <http://www.R-project.org>. ISBN 3-900051-07-0.

]8[

L. Torgo: *Introducao a programacao em R..* 2006.

]9[

W. N. Venables, D. M. Smith, R Development Core Team: *An Introduction to R (Notes on R: A Programming Environment for Data Analysis and Graphics)*. The R Foundation for Statistical Computing, 2006. URL <http://cran.r-project.org/doc/manuals/R-intro.pdf>.

]10[

W.P. Bowen, J.C. Jerman: “Nonlinear regression using spreadsheets”, *Trends in Pharmacological Sciences*, pp. 413-417, 1995.

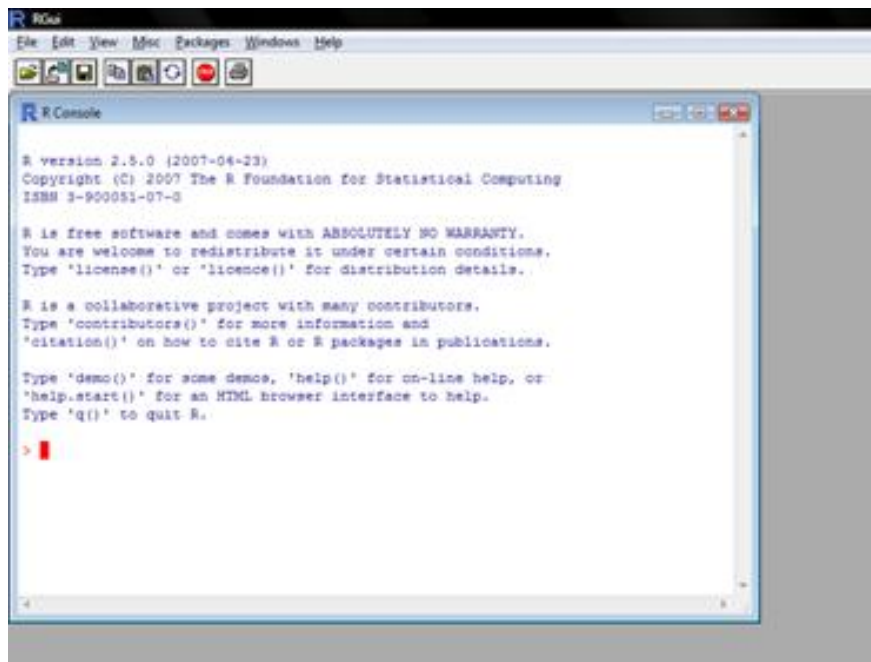


Figura 6: This is a caption

Figure 1 Aspecto da janela *R Console*.

Na consola do R escrever (fazendo Enter no fim),

```
> dados<-read.table(file="G:/MyArticles/ArtigosRegressaoNaoLinearExcel/RegressaoNaoLinear.txt",header=T,sep="\t")
```

para “carregar” a informação contida no ficheiro *RegressaoNaoLinear.txt* para o objecto *dados*. (atente-se na notação `<-`). Os parâmetros *header* e *sep* permitem considerar os rótulos dos dados e o tipo de ficheiro de texto (neste caso separado por tabulações), respectivamente. Por sinal, a quantidade de dados envolvida poderia muito bem ser introduzida directamente no R através de

```
> CO2<-c(141,172,181,227,309,414,641, 936)
```

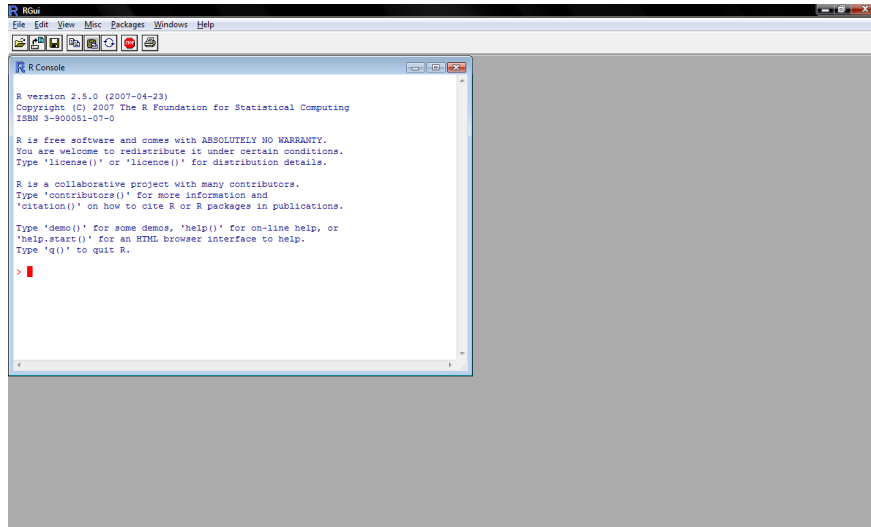


Figura 7: This is a caption

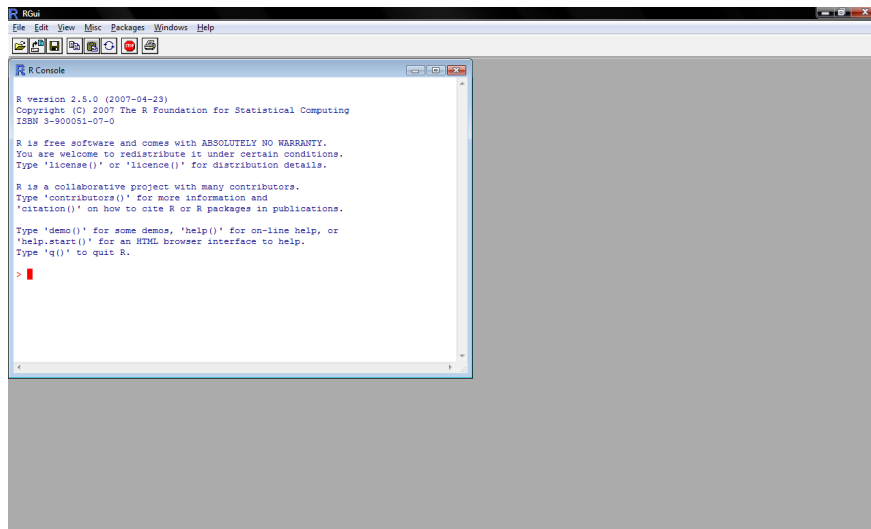


Figura 8: This is a caption

```
> Prop<-c(0.25,0.34,0.34,0.68,0.85,0.99,0.98,0.99)
```

em que `c(...)` permite concatenar um conjunto de dados num objecto (neste caso, CO2 e Prop) . Podemos visualizar a informação usando a função `plot`

```
> plot(Prop~CO2, las=1,xlim=c(0,1000),ylim=c(0,1),pch=16,cex=
1.5, xlab="CO2 (ppm)", ylab="Proporção de respostas correctas")
```

em que `Prop~CO2` descreve o “modelo” que se pretende representar (neste caso Prop em função de CO2) e cujos restantes parâmetros especificam as várias características do gráfico (`las` para orientação dos rótulos do eixo dos *yy*, `xlim` e `ylim` para determinar os limites dos eixos, `pch` para definir o símbolo usado, `cex` para aumentar o tamanho relativo dos caracteres, e `xlab` e `ylab` para “legendar” os eixos) (Figura a). Para mais

informações acerca desta (ou qualquer outra) função pode fazer-se

```
> ?plot
```

(ou então procurar com `>help.search("termo")` ou, ainda, usar a opção `Help>Apropos...`). Usando o botão direito do rato sobre o gráfico, é possível copiar a figura (como metaficheiro `Windows®` por exemplo) para inclusão num ficheiro do `Microsoft Word®`.

Admita-se (por razões científicas) que um modelo “dose-resposta” (logístico modificado)

$$P_i = \frac{1}{1 + \exp(B(T_i - \log(x)))}$$

em que  $P_i$  é a proporção de respostas positivas do provador  $i$ ,  $\log(x)$  é o logaritmo do valor do estímulo  $x$ ,  $B$  é o “declive” e  $T$  é o limiar para o provador  $i$  (em  $\log(x)$ ), e adequado para descrever os dados. Para ajustar o modelo basta fazer

```
> ajuste<-nls(Prop~(1/3+exp(B*(T-log10(CO2))))/(1+exp(B*(T-log10(CO2)))),  
data=data.frame(CO2=CO2,Prop=Prop),start=list(B=-10,T=2.5))
```

em que o objecto `ajuste` inclua os resultados duma regressão não-linear (através da função `nls`)

```
]C[
```

```
]C[
```

Outras funções (muito) usadas são `lm` e `aov` para regressão linear (i.e. modelos lineares) e análise de variância, respectivamente. do modelo

```
]D[
```

```
]D[
```

A definição dos modelos em S está minuciosamente descrita em

```
] > Prop~(1/3+exp(B*(T-log10(CO2))))/(1+exp(B*(T-log10(CO2)))) > considerandoosdados(indicadosparaop...
```

```
]E[
```

Neste caso, considerou-se que o utilizador introduziu directamente no sistema os dados ao invés de “carregar” o ficheiro dos dados (sendo que bastaria apenas indicar `data=dados`, no comando. ) e as estimativas iniciais dos parâmetros do modelo descritas em `start` (através duma `list`). Os resultados do ajuste (Figura ) podem ser consultados fazendo

```
> summary(ajuste)
```

Figure 2 Diagramas de dispersão da proporção de respostas vs. concentração de CO<sub>2</sub>: dados originais (a); modelo “dose-resposta” (logístico modificado) (b); modelo “dose-resposta” com intervalos de 95% de confiança (c); e modelo logístico (ver texto principal) (d).

É possível ilustrar o ajuste, obtendo primeiro as estimativas da variável `Prop` (através da função `predict`) e “desenhando-as”, em seguida, sobre o gráfico entretanto obtido (Figura ou seja

```
> PropEsp<-predict(ajuste,newdata=list(CO2=seq(0,1000,length=100)),se.fit=T)
> lines(PropEsp~seq(0,1000,length=100),lwd=2,col=6)
```

A informacao contida no objecto ajuste permite obter as estimativas para 100 valores de CO2 entre 0 e 1000 (dai o parametro seq(0,1000,length=100) no comando acima) assim como os respectivos erros-padrao (se.fit=T) que serao guardadas no objecto designado PropEsp. A adicao duma linha espessa e magenta (lwd=2 e col=6) a Figura a e relativamente facil (Figura b).

Figure 3 Aspecto da execucao de comandos e respectivos resultados na janela *R Console*.

Podem, ainda, obter-se e “desenhar-se” os intervalos de confianca (upIC e loIC) que se calculam atraves de  $y \pm t_{\alpha/2} se(Y)$ , recorrendo as funcoes qt (para obter os valores “teoricos” de t-Student e cujos parametros sao  $1-\alpha/2$ , neste caso 0.975, e os graus de liberdade do erro, que se obtêm de summary(ajuste)\$df

```
]2[
), summary(ajuste)$sigma para obter o erro-padrao das estimativas e lines, respectivamente (Figura
```

```
> upIC<-PropEsp+qt(.975,summary(ajuste)$df
```

```
]2[
```

```
)*summary(ajuste)$sigma
```

```
> loIC<-PropEsp-qt(.975,summary(ajuste)$df
```

```
]2[
```

```
)*summary(ajuste)$sigma
```

```
> lines(upIC~seq(0,1000,length=100),col=4)
```

```
> lines(loIC~seq(0,1000,length=100),col=4)
```

O modo mais simples de “testar” a bondade do ajuste é representar graficamente os dados e o modelo ajustado (bem como os respectivos intervalos de confiança), de modo a verificar (visualmente) se os parâmetros obtidos numericamente descrevem, de facto, a relação entre variáveis (Figura c). Por outro lado, a análise gráfica dos resíduos, e.g. resíduos *vs.* valores observados de  $X$ , permite “verificar” se o modelo é adequado (os resíduos representam apenas o erro experimental se não apresentam tendências ou padrões). Sendo assim, obter os resíduos através de

```
> residuos<-summary(ajuste)$resid
```

e, em seguida, representar esse gráfico em conjunto com o *normal probability plot* dos resíduos (Figura )

```
> op<-par(mfrow=c(2,1),pty="s",las=1)
```

```
> plot(residuos~predict(ajuste),pch=16,col=3,xlab="E{Proporção de respostas
correctas}",ylab="Resíduos")
```

```
> abline(h=0)
```

```
> qqnorm(residuos,pch=16,col=2,xlab="Quantis da Distribuição Normal",
ylab="Quantis Amostrais",main="")
```

```
> qqline(residuos)
```

```
> par(op)
```

O comando `par` permite estabelecer o “esquema” da figura, neste caso, os gráficos ficam “arranjados” em 2 linhas por 1 coluna – `mfrow=c(2,1)` –, assim como definir a respectiva forma (quadrada, através de `pty=“s”` - “s” de “square”). Ao gráfico dos resíduos *vs.* valores esperados de `Prop` (através de `plot`) acrescentámos uma linha horizontal ao nível de zero (através de `abline`) e, por fim, desenhamos o *normal probability plot* dos resíduos (com `qqnorm` e `qqline`).

Figure 4 Analise grafica de residuos: residuos *vs.* valores esperados (esq.); e *normal probability plot* dos residuos (dir.).

E possível testar se os resíduos se distribuem normalmente através, por exemplo, de

```
> shapiro.test(residuos)
```

O resultado deste comando é

Shapiro-Wilk normality test

data: residuos

W = 0.936, p-value = 0.5719

O teste de sequências (ou “runs test” nos manuais anglofonos) permite testar, de forma simples e robusta, se os pontos (dados) diferem sistematicamente da curva ajustada (modelo) complementando a informação do diagrama dos resíduos *vs.* valores observados de  $X$ .

*[alertaparaareduziapotenciaestatisticadotestesbaseados > emsequencias.NoR, seranecessariocarregaralguns]*

(Figura d)

```
]F[
```

```
]F[
```

Esta figura foi obtida através de

```
> plot(Prop~CO2,las=1,xlim=c(0,1000), ylim=c(0,1),pch=16,cex=1.5, xlab=“CO2  
(ppm)”,ylab=“Propor
```

```
ção de respostas correctas“)
```

```
> PropEsper<-predict(ajustes,newdata= list(CO2=seq(0,1000,length=100)),se.fit=T)
```

```
> lines(PropEsper~seq(0,1000,length= 100),lwd=2,col=6)
```

```
> uppIC<-PropEsper+qt(.975,summary( ajustes)$df
```

```
]2[
```

```
)*summary(ajustes)$sigm
```

```
> lowIC<-PropEsper-qt(.975,summary( ajustes)$df
```

```
]2[
```

```
)*summary(ajustes)$sigm
```

```
> lines(uppIC~seq(0,1000,length=100), col=4)
```

```
> lines(lowIC~seq(0,1000,length=100), col=4)
, de formulação mais generalizada do que o modelo “dose-resposta” anterior, através
de
> ajustes<-nls(Prop~C/(1+A*exp(-B*CO2)),data=data.frame(CO2=CO2,Prop=Prop),
start=list(A=10,B=.0025,C=1))
> summary(ajustes)
```

os resultados do teste de F obtêm-se facilmente através de

```
> anova(ajuste,ajustes)
```

Os passos descritos anteriormente, i.e. uma sessão de trabalho (Figura ), podem (e, na minha opinião, devem) ser “guardados”. Para isso, criar um ficheiro de *script* (através de File>New script), de facto um ficheiro texto, onde são “guardados” os comandos. Este ficheiro pode ser carregado noutra sessão de trabalho (através de File>Open script...) e as análises e gráficos podem ser repetidos n-vezes (janela *R Editor*). O *script* do exemplo descrito aqui pode ser obtido no sítio electrónico do autor.

Para terminar uma sessão de trabalho, primeiro “limpar” da memória os objectos criados (com a função `rm`) e, depois, encerrar o programa (geralmente não guardo uma “imagem” do ambiente de trabalho):

```
> rm(list=ls()); q()
```

Figure 5 Aspecto duma sessão de trabalho no R.

Consideracoes finais

As “dificuldades estatísticas” relacionadas com a utilizacao do Excel(r) da Microsoft (e da respectiva ferramenta Solver(r)) para analise de regressao nao-linear

*]limitamasuutilizacaoaoscasos” mais > simples”ouparae feitos pedagogicos.Emsituacoes” profissionais” > de*

Embora as características do R tornem (mais) difícil a aprendizagem, o seu carácter gratuito, o facto de ser uma ferramenta bastante poderosa com boas capacidades ao nível da programação e um conjunto bastante vasto (e em constante crescimento) de *\*packages\** que acrescentam bastantes potencialidades estatísticas e gráficas, o crescente interesse de utilizadores das mais variadas formações (que se reúnem anualmente desde 2004 nas conferências useR! – em 2011 a useR!2011 decorrerá no Reino Unido, *\*cf\**. <http://www.warwick.ac.uk/statsdept/useR-2011/>) e a disponibilidade dos promotores (R Development Core Team) e da “comunidade” de utilizadores, mais ou menos avançados, ajudam (e muito) esse processo. Os vários manuais (<http://cran.r-project.org/manuals.html>), a página wiki (<http://wiki.r-project.org/>) e a *\*mailing list\** de ajuda (<http://stat.ethz.ch/mailman/listinfo/r-help>) permitem ultrapassar a (esmagadora) maioria das dificuldades surgidas durante a utilização do R.

∫ Agradecimentos ∫ ∫ Este artigo muito beneficiou dos comentários, correções e sugestões ∫ dos colegas e leitores. ∫ ∫ Referências ∫ ∫ [

]1[

A. M. Brown: “A step-by-step guide to non-linear regression analysis of experimental data using a Microsoft Excel spreadsheet”, *Computer methods and Programs in Biomedicine*, pp. 191-200, 2001.

]2[

John M. Chambers, Trevor J. Hastie: *Statistical models in S*. Chapman & Hall, 1992.



]3[

W.J. Conover: *Practical nonparametric statistics*. John Wiley & Sons. Inc., 1999.

]4[

M.J. Crawley: *The R book*. Wiley, 2007. URL <http://www.google.com/books?id=8D4HVx0apZQC>.

]5[

Peter Dalgaard: *Introductory statistics with R*. Springer-Verlag New York Inc., 2002.

]6[

E. Esteves: *Regressao nao-linear utilizando a ferramenta Solver(r) do Microsoft Excel(r)*. 2010. URL <http://w3.ualg.pt/~eesteves..>

]7[

R Development Core Team: *R: A Language and Environment for Statistical Computing*. 2007. URL <http://www.R-project.org>. ISBN 3-900051-07-0.

]8[

L. Torgo: *Introducao a programacao em R..* 2006.

]9[

W. N. Venables, D. M. Smith, R Development Core Team: *An Introduction to R (Notes on R: A Programming Environment for Data Analysis and Graphics)*. The R Foundation for Statistical Computing, 2006. URL <http://cran.r-project.org/doc/manuals/R-intro.pdf>.

]10[

W.P. Bowen, J.C. Jerman: “Nonlinear regression using spreadsheets”, *Trends in Pharmacological Sciences*, pp. 413-417, 1995.

Figure 1 Aspecto da janela *R Console*.

Na consola do R escrever (fazendo Enter no fim),

```
> dados<-read.table(file="G:/MyArticles/ArtigosRegressaoNaoLinearExcel/RegressaoNaoLinear.txt",header=T,sep="\t")
```

para “carregar” a informacao contida no ficheiro *RegressaoNaoLinear.txt* para o objecto *dados*. (atente-se na notacao <-). Os parametros *header* e *sep* permitem considerar os rotulos dos dados e o tipo de ficheiro de texto (neste caso separado por tabulacoes), respectivamente. Por sinal, a quantidade de dados envolvida poderia muito bem ser introduzida directamente no R atraves de

```
> CO2<-c(141,172,181,227,309,414,641, 936)
```

```
> Prop<-c(0.25,0.34,0.34,0.68,0.85,0.99,0.98,0.99)
```

em que `c(...)` permite concatenar um conjunto de dados num objecto (neste caso, `CO2` e `Prop`) . Podemos visualizar a informacao usando a funcao `plot`

```
> plot(Prop~CO2, las=1,xlim=c(0,1000),ylim=c(0,1),pch=16,cex=
1.5, xlab="CO2 (ppm)", ylab="Proporcao de respostas correctas")
```

em que `Prop~CO2` descreve o “modelo” que se pretende representar (neste caso `Prop` em funcao de `CO2`) e cujos restantes parametros especificam as varias caracteristicas do grafico (las para orientacao dos rotulos do eixo dos *yy*, `xlim` e `ylim` para determinar os limites dos eixos, `pch` para definir o simbolo usado, `cex` para aumentar o tamanho relativo dos caracteres, e `xlab` e `ylab` para “legendar” os eixos) (Figura a). Para mais informacoes acerca desta (ou qualquer outra) funcao pode fazer-se

```
> ?plot
```

(ou entao procurar com `>help.search(“termo”)` ou, ainda, usar a opcao `Help>`

Apropos... Usando o botão direito do rato sobre o gráfico, é possível copiar a figura (como meta ficheiro `Windows®` por exemplo) para inclusão num ficheiro do `Microsoft Word®`.

Admita-se (por razões científicas) que um modelo “dose-resposta” (logístico modificado)

$$P_i = \frac{1}{3} \frac{\exp(B(T_i - \log(x)))}{1 + \exp(B(T_i - \log(x)))}$$

em que  $P_i$  e a proporcao de respostas positivas do provador  $i$  ,  $\log(x)$  e o logaritmo do valor do estimulo  $x$  ,  $B$  e o “declive” e  $T$  e o limiar para o provador  $i$  (em  $\log(x)$ ), e adequado para descrever os dados. Para ajustar o modelo basta fazer

```
> ajuste<-nls(Prop~(1/3+exp(B*(T-log10(CO2))))/(1+exp(B*(T-
log10(CO2))))),
```

```
data=data.frame(CO2=CO2,Prop=Prop),start=list(B=-10,T=2.5))
```

em que o objecto `ajuste` incluire os resultados duma regressao nao-linear (atraves da funcao `nls`)

```
]C[
```

```
]C[
```

Outras funcoes (muito) usadas sao `lm` e `aov` para regressao linear (i.e. modelos lineares) e analise de variancia, respectivamente. do modelo

```
]D[
```

```
]D[
```

A definicao dos modelos em S esta minuciosamente descrita em

```
] > Prop~(1/3+exp(B*(T-log10(CO2))))/(1+exp(B*(T-log10(CO2)))) > considerandoosdados(indicadospor
```

```
]E[
```

Neste caso, considerou-se que o utilizador introduziu directamente no sistema os dados ao inves de “carregar” o ficheiro dos dados (sendo que bastaria apenas indicar `data=dados`, no comando). ) e as estimativas iniciais dos parametros do modelo descritas em `start` (atraves duma `list`). Os resultados do ajuste (Figura ) podem ser consultados fazendo

```
> summary(ajuste)
```

Figure 2 Diagramas de dispersao da proporcao de respostas vs. concentracao de CO2: dados originais (a); modelo “dose-resposta” (logistico modificado) (b); modelo “dose-resposta” com intervalos de 95% de confianca (c); e modelo logistico (ver texto principal) (d).

E possivel ilustrar o ajuste, obtendo primeiro as estimativas da variavel `Prop` (atraves da funcao `predict`) e “desenhando-as”, em seguida, sobre o grafico entretanto obtido (Figura ou seja

```
> PropEsp<-predict(ajuste,newdata=list(CO2=seq(0,1000,length=100)),se.fit=T)
> lines(PropEsp~seq(0,1000,length=100),lwd=2,col=6)
```

A informacao contida no objecto `ajuste` permite obter as estimativas para 100 valores de CO2 entre 0 e 1000 (dai o parametro `seq(1,1000,length=100)` no comando acima) assim como os respectivos erros-padrao (`se.fit=T`) que serao guardadas no objecto designado `PropEsp`. A adicao duma linha espessa e magenta (`lwd=2` e `col=6`) a Figura a e relativamente facil (Figura b).

Figure 3 Aspecto da execucao de comandos e respectivos resultados na janela *R Console*.

Podem, ainda, obter-se e “desenhar-se” os intervalos de confianca (`upIC` e `loIC`) que se calculam atraves de  $y \pm t[?]se(Y)$ , recorrendo as funcoes `qt` (para obter os valores “teoricos” de t-Student e cujos parametros sao  $1[?]-[?]a[?]/[?]2$ , neste caso 0.975, e os graus de liberdade do erro, que se obtêm de `summary(ajuste)$df`

```
]2[
```

), `summary(ajuste)$sigma` para obter o erro-padrão das estimativas e `lines`, respectivamente (Figura

```
> upIC<-PropEsp+qt(.975,summary(ajuste)$df
```

```
]2[
```

```
)*summary(ajuste)$sigma
```

```
> loIC<-PropEsp-qt(.975,summary(ajuste)$df
```

```
]2[
```

```
)*summary(ajuste)$sigma
```

```
> lines(upIC~seq(0,1000,length=100),col=4)
```

```
> lines(loIC~seq(0,1000,length=100),col=4)
```

O modo mais simples de “testar” a bondade do ajuste é representar graficamente os dados e o modelo ajustado (bem como os respectivos intervalos

de confiança), de modo a verificar (visualmente) se os parâmetros obtidos numericamente descrevem, de facto, a relação entre variáveis (Figura c). Por outro lado, a análise gráfica dos resíduos, e.g. resíduos *vs.* valores observados de  $X$ , permite “verificar” se o modelo é adequado (os resíduos representam apenas o erro experimental se não apresentam tendências ou padrões). Sendo assim, obter os resíduos através de

```
> residuos<-summary(ajuste)$resid
e, em seguida, representar esse gráfico em conjunto com o normal probability plot dos resíduos (Figura )
> op<-par(mfrow=c(2,1),pty="s",las=1)
> plot(residuos~predict(ajuste),pch=16,col=3,xlab="E{Proporção de
respostas
correctas}",ylab="Resíduos")
> abline(h=0)
> qqnorm(residuos,pch=16,col=2,xlab="Quantis da Distribuição Normal",
ylab="Quantis Amostrais",main="")
> qqline(residuos)
> par(op)
```

O comando `par` permite estabelecer o “esquema” da figura, neste caso, os gráficos ficam “arranjados” em 2 linhas por 1 coluna – `mfrow=c(2,1)` –, assim como definir a respectiva forma (quadrada, através de `pty="s"` - “s” de “square”). Ao gráfico dos resíduos *vs.* valores esperados de Prop (através de `plot`) acrescentámos uma linha horizontal ao nível de zero (através de `abline`) e, por fim, desenhamos o *normal probability plot* dos resíduos (com `qqnorm` e `qqline`).

Figure 4 Analise grafica de residuos: residuos *vs.* valores esperados (esq.); e *normal probability plot* dos residuos (dir.).

E possível testar se os resíduos se distribuem normalmente através, por exemplo, de

```
> shapiro.test(residuos)
O resultado deste comando e
Shapiro-Wilk normality test
data: residuos
W = 0.936, p-value = 0.5719
```

O teste de sequências (ou “runs test” nos manuais anglofonos) permite testar, de forma simples e robusta, se os pontos (dados) diferem sistematicamente da curva ajustada (modelo) complementando a informação do diagrama dos resíduos *vs.* valores observados de  $X$ .

[alertaparaareduziapotenciaestatisticadotestesbaseados > emsequencias.NoR, seranecessariocarregar]

(Figura d)

]F[

]F[

Esta figura foi obtida através de

```
> plot(Prop~CO2,las=1,xlim=c(0,1000), ylim=c(0,1),pch=16,cex=1.5,
xlab="CO2 (ppm)",ylab="Propor
ção de respostas correctas")
> PropEsper<-predict(ajustes,newdata= list(CO2=seq(0,1000,length=100)),se.fit=T)
> lines(PropEsper~seq(0,1000,length= 100),lwd=2,col=6)
> uppIC<-PropEsper+qt(.975,summary(ajustes)$df
```

]2[

```
)*summary(ajustes)$sigm
> lowIC<-PropEsper-qt(.975,summary(ajustes)$df
```

]2[

```
)*summary(ajustes)$sigm
> lines(uppIC~seq(0,1000,length=100), col=4)
> lines(lowIC~seq(0,1000,length=100), col=4)
```

, de formulação mais generalizada do que o modelo “dose-resposta” anterior, através de

```
> ajustes<-nls(Prop~C/(1+A*exp(-B*CO2)),data=data.frame(CO2=CO2,Prop=Prop),
start=list(A=10,B=.0025,C=1))
> summary(ajustes)
```

os resultados do teste de F obtêm-se facilmente através de

```
> anova(ajuste,ajustes)
```

Os passos descritos anteriormente, i.e. uma sessão de trabalho (Figura ), podem (e, na minha opinião, devem) ser “guardados”. Para isso, criar um ficheiro de *script* (através de File>New script), de facto um ficheiro texto, onde são “guardados” os comandos. Este ficheiro pode ser carregado noutra sessão de trabalho (através de File>Open script...) e as análises e gráficos podem ser repetidos n-vezes (janela *R Editor*). O *script* do exemplo descrito aqui pode ser obtido no sítio electrónico do autor.

Para terminar uma sessão de trabalho, primeiro “limpar” da memória os objectos criados (com a função `rm`) e, depois, encerrar o programa (geralmente não guardo uma “imagem” do ambiente de trabalho):

```
> rm(list=ls()); q()
```

Figure 5 Aspecto duma sessão de trabalho no R.

## Consideracoes finais

As “dificuldades estatísticas” relacionadas com a utilizacao do Excel(r) da Microsoft (e da respectiva ferramenta Solver(r)) para analise de regressao nao-linear

]limitamasuautilizacaoaoscasos” mais > simples” ou para feitos pedagogicos. Em situacoes” profissionais

Embora as características do R tornem (mais) difícil a aprendizagem, o seu carácter gratuito, o facto de ser uma ferramenta bastante poderosa com boas capacidades ao nível da programação e um conjunto bastante vasto (e em constante crescimento) de \*packages\* que acrescentam bastantes potencialidades estatísticas e gráficas, o crescente interesse de utilizadores das mais variadas formações (que se reúnem anualmente desde 2004 nas conferências useR! – em 2011 a useR!2011 decorrerá no Reino Unido, \*cf\*. <http://www.warwick.ac.uk/statsdept/useR-2011/>) e a disponibilidade dos promotores (R Development Core Team) e da “comunidade” de utilizadores, mais ou menos avançados, ajudam (e muito) esse processo. Os vários manuais (<http://cran.r-project.org/manuals.html>), a página wiki (<http://wiki.r-project.org/>) e a \*mailing list\* de ajuda (<http://stat.ethz.ch/mailman/listinfo/r-help>) permitem ultrapassar a (esmagadora) maioria das dificuldades surgidas durante a utilização do R.

¿ Agradecimentos ¿ Este artigo muito beneficiou dos comentários, correções e sugestões ¿ dos colegas e leitores. ¿ ¿ Referências ¿ ¿ [

]1[

A. M. Brown: “A step-by-step guide to non-linear regression analysis of experimental data using a Microsoft Excel spreadsheet”, *Computer methods and Programs in Biomedicine*, pp. 191-200, 2001.

]2[

John M. Chambers, Trevor J. Hastie: *Statistical models in S*. Chapman & Hall, 1992.

]3[

W.J. Conover: *Practical nonparametric statistics*. John Wiley & Sons. Inc., 1999.

]4[

M.J. Crawley: *The R book*. Wiley, 2007. URL <http://www.google.com/books?id=8D4HVx0apZQC>.

]5[

Peter Dalgaard: *Introductory statistics with R*. Springer-Verlag New York Inc., 2002.

]6[

E. Esteves: *Regressao nao-linear utilizando a ferramenta Solver(r) do Microsoft Excel(r)*. 2010. URL <http://w3.ualg.pt/~eesteves..>

]7[

R Development Core Team: *R: A Language and Environment for Statistical Computing*. 2007. URL <http://www.R-project.org>. ISBN 3-900051-07-0.

]8[

L. Torgo: *Introducao a programacao em R..* 2006.

]9[

W. N. Venables, D. M. Smith, R Development Core Team: *An Introduction to R (Notes on R: A Programming Environment for Data Analysis and Graphics)*. The R Foundation for Statistical Computing, 2006. URL <http://cran.r-project.org/doc/manuals/R-intro.pdf>.

]10[

W.P. Bowen, J.C. Jerman: “Nonlinear regression using spreadsheets”, *Trends in Pharmacological Sciences*, pp. 413-417, 1995.

## Referências

Bowen, W. P. and Jerman, J. C. (1995) “Nonlinear regression using spreadsheets”, *Trends in Pharmacological Sciences*, 16, pp. 413–417.

Brown, A. M. (2001) “A step-by-step guide to non-linear regression analysis of experimental data using a Microsoft Excel spreadsheet”, *Computer methods and Programs in Biomedicine*, 65, pp. 191–200.

Chambers, J., Hastie, T. and Pregibon, D. (1990) “Statistical Models in S”, in *Compstat*. Physica-Verlag HD, pp. 317–321. doi: 10.1007/978-3-642-50096-1\_48.

Crawley, M. J. (2007) *The R book*. NY: Wiley. Available at: <http://www.google.com/books?id=8D4HVx0apZQC>.

Dalgaard, P. (2002) *Introductory statistics with R*. New York: Springer-Verlag New York Inc.

Esteves, E. (2011) “Regressão não-linear utilizando a ferramenta Solver® do Microsoft Excel®”. Instituto Superior de Engenharia, Universidade do Algarve, Faro Portugal [disponível em <http://w3.ualg.pt/~eesteves>]. Available at: <http://w3.ualg.pt/~eesteves>.

Torgo, L. (2006) *Introdução à programação em R..* Porto: Faculdade de Economia da Universidade do Porto.

Venables, W. N., Smith, D. M. and Team, R. D. C. (2006) *An Introduction to R (Notes on R: A Programming Environment for Data Analysis and Graphics)*. Vienna, Austria: The R Foundation for Statistical Computing. Available at: <http://cran.r-project.org/doc/manuals/R-intro.pdf>.

R Core Team and R Development Core Team (2017) *R: A language and environment for statistical computing..* Vienna, Austria: R Foundation for Statistical Computing. Available at: <http://www.r-project.org>.