Solving Generalized Linear Models with Spreadsheets

Antonio Fernando de Castro Vieira
Departamento de Engenharia Industrial, PUC-Rio
afcv@ind.puc-rio.br

Eugenio Kahn Epprecht
Departamento de Engenharia Industrial, PUC-Rio
E-mail: eke@ind.puc-rio.br

Abstract
Design and analysis of experiments are extensively used in quality improvement efforts, for identifying the effects of controllable variables (factors) over response variables associated to the quality level of a product. When the effects are not simply additive and/or the response variance is not homogeneous (as is the case of many industrial applications), Generalized Linear Models (GLMs) constitute a widely suitable approach to the analysis. Their application is however hindered if specialized software for solution is not available – a situation which leads in general to the use of simpler but inappropriate models, and consequently to invalid results and conclusions. This paper shows how GLMs can be solved using spreadsheet software. This can fit the needs of occasional users, needing to solve not very large problems; it can also be advantageous for teaching and training purposes.
Key Words: Design of Experiments; Quality Improvement; Regression Analysis; Generalized Linear Models; Spreadsheet Applications.

Introduction

Statistical design of experiments (DOE) is extensively used in quality improvement efforts, with the aim of identifying the relationship between controllable variables (*factors*) and response variables associated to the quality level of a product. The model most commonly used to represent the functional relationship between the controllable variables x_i and the response variable y (considering here the case of a univariate response) is the classic linear additive model:

$$y = \mu(x) + e \quad (1)$$

where:

$$\mu(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k \quad (2)$$

represents the systematic part of the model, i.e., the mean level of the response for each value of the vector of controllable variables $\mathbf{x}' = (x_1, x_2, \ldots, x_k)$, and e, called the *error*, or *noise*, is the random component that represents the inherent variability of the response around the mean level. Note that Eq. (2) can accommodate models that are nonlinear in the original variables, by simply writing functions of one or more variables as new variables (for example, representing x_1x_2 by x_3, or x_1^3 by x_4), provided that the resulting expression is linear in the parameters.

Traditionally, the model parameters $\beta_1, \beta_2, \ldots \beta_k$ are estimated using the ordinary least squares method (OLS), which assumes linearity in the parameters and normality and homogeneity of the variance of the response, that is, $e \sim N(0, \sigma^2)$.

Industrial applications very often involve models that do not satisfy these conditions, for instance binomial models for fraction defective, Poisson models for counts of defects or gamma models used in life-testing.

The usual approach in such cases has been to work with a transformed response: Bisgaard and Fuller (1994) present two examples: one for binomial responses and other for Poisson responses. Box and Fung (1995) analyzed designed experiments for life testing. The drawback of this approach is that, with a single transformation, one cannot always obtain at the same time an additive model and a normal response with constant variance. An alternative approach is offered by the Generalized Linear Models (GLM), which do not require these conditions.

In a GLM the response may follow any probability distribution from the exponential family (for a description of the exponential family of distributions, see, for instance, McCullough and Nelder, 1989). The normal, exponential, gamma, Poisson and binomial distributions, for example, belong to this family. For different values (*levels*) of the factors, the response – while possessing similar probability distributions – may have different parameter values. Moreover, the functional relationship between the controllable variables and the response is not required to be linear as in Eq. (2), but just given by a (monotone and differentiable) *link function* $g(\cdot)$: a function of the mean response whose value equals a linear combination of the controllable variables, that is:

$$\eta = g(\mu) = \mathbf{x}'\beta \qquad (3)$$

where β is the vector of parameters of the model.

In other words, the model for the response differs from Eq. (2) in that the left-hand side is a function $g(\mu)$ of the mean response, instead of the mean response itself.

A detailed account of GLMs can be found in (McCullough and Nelder, 1989). A simpler introduction to the subject is offered in (Dobson, 1990). Illustrative examples with bino-

mial, Poisson and gamma responses, in which the GLMs results outperformed the results of the analysis of the transformed response, are presented in Hamada and Nelder (1997), Myers and Montgomery (1997) and Lewis *et al.* (2001).

The solution of GLMs requires finding the estimates of the parameters β_i that maximize likelihood of the set of observations. This is accomplished by numerical search procedures involving a certain amount of computation. Routines for solution of GLMs are found in some statistical packages, such as GLIM and GenStat. The non-availability of this kind of software may hinder the employment of GLMs, leading eventually to the use of less appropriate models.

This paper shows how GLMs can be solved with electronic spreadsheets, being intended for occasional users of GLMs, for small problems. We also trust that solving GLMs with spreadsheets can be useful for teaching and training purposes, by rendering apparent the mechanics of solution. A model with Poisson response is used here as an illustration, but the approach is general, and applicable to other models, requiring only trivial modifications.

The next two sections describe the model and the solution algorithm. The spreadsheet implementation of the algorithm is then illustrated by means of a numerical example.

Generalized Linear Model with Poisson Response and Logarithmic Link Function

Denote by **y** the vector of the n observed values of the Poisson-distributed response of an unreplicated 2^k factorial design. The index i (i=1,2,..., n) refers to the combination of levels of the factors (*run*). The mass probability function of each element y_i of **y** is:

$$f_Y(y_i, \mu_i) = \frac{e^{\mu_i} \mu_i^{y_i}}{y_i!} \tag{4}$$

Let us consider the multiplicative model for the mean, in the form:

$$\mu_i = exp\left(\beta_0 + \sum_{j=1}^{k} x_{ij} \beta_j \right) \tag{5}$$

where x_{ij} is the value of the covariate x_j in the i-th run; thus, p=k+1 is the number of model parameters.

Recalling that the relationship between the mean level of the response and the vector **x** should be given by a *link function* $\eta = g(\mu) = \mathbf{x}' \beta$, let us define x_{i0}=1, for every i=1, 2, ..., n. This puts (5) into the form μ_i=exp($\mathbf{x}_i' \beta$), where $\mathbf{x}_i' = (1, x_{i1}, x_{i2}, ..., x_{in})$, corresponding therefore to the link function:

$$\eta = ln\mu = \mathbf{x}' \beta \tag{6}$$

The likelihood function of n observations from a Poisson distribution is:

$$L(\mathbf{y},\mu)=\prod_{i=1}^{n} f_Y(y_i,\mu_i)=\prod_{i=1}^{n}\frac{e^{\mu_i}\mu_i{}^{y_i}}{y_i!} \tag{7}$$

Taking natural logarithms, the log-likelihood function is found to be:

$$l(\mathbf{y},\mu)=ln\,L(\mathbf{y},\mu)=\sum_{i=1}^{n}\left[y_i\,ln\,\mu_i-\mu_i-ln(y_i!)\right] \tag{8}$$

The reader non-familiar with GLMs is referred to (McCullough and Nelder, 1989) or to (Dobson, 1990).

Algorithm for Parameter Estimation

The algorithm for obtaining the maximum likelihood estimate **b** for the vector of parameters β_i is described below. This algorithm is called IRSL – *iterative reweighted least squares* (see, for instance, McCullough and Nelder, 1989, or Dobson, 1990). It makes use of the equation:

$$\mathbf{b}^{(m+1)}=\left(\mathbf{X}'\mathbf{W}^{(m)}\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{W}^{(m)}\mathbf{z}^{(m)} \tag{9}$$

where **b** is the estimated parameter vector

$$\mathbf{b}^{(m)}=\begin{bmatrix}\hat{\beta}_1^{(m)}\\ \hat{\beta}_2^{(m)}\\ \vdots\\ \hat{\beta}_k^{(m)}\end{bmatrix} \tag{10}$$

X is the matrix of values of the controllable variables, in the form

$$\mathbf{X}=\begin{bmatrix}1 & x_{11} & \cdots & x_{1k}\\ 1 & x_{21} & \cdots & x_{2k}\\ \vdots & \vdots & \cdots & \vdots\\ 1 & x_{n1} & \cdots & x_{nk}\end{bmatrix} \tag{11}$$

W is a matrix of weights

$$\mathbf{W}^{(m)}=\begin{bmatrix}w_{11}^{(m)} & 0 & \cdots & 0\\ 0 & w_{22}^{(m)} & \cdots & 0\\ \vdots & \vdots & \cdots & \vdots\\ 0 & 0 & \cdots & w_{nn}^{(m)}\end{bmatrix} \tag{12}$$

whose diagonal elements $w_{ii}^{(m)}$ are given by:

$$w_{ii}^{(m)} = \frac{1}{\text{var}(Y_i)}\left(\left(\frac{\partial \mu_i}{\partial \eta_i}\right)^{(m)}\right)^2 \qquad i = 1, 2, \cdots, n \tag{13}$$

and $\mathbf{z}$ is the vector of adjusted variables

$$\mathbf{z}^{(m)} = \begin{bmatrix} z_1^{(m)} \\ z_2^{(m)} \\ \vdots \\ z_n^{(m)} \end{bmatrix} \tag{14}$$

with elements given by

$$z_i^{(m)} = \hat{\eta}_i^{(m)} + \left(y_i - \hat{\mu}_i^{(m)}\right)\left(\frac{\partial \eta_i}{\partial \mu_i}\right)^{(m)} \qquad i = 1, 2, \cdots, n \tag{15}$$

Examining Eq. (15), recall that η and μ depend on β through (3). This shows that Eq. (9) is recursive. In each iteration, the parameter vector estimate $\mathbf{b}^{(m+1)}$ is computed as a function of the previous estimate $\mathbf{b}^{(m)}$.

In our experience, when given a good initial solution, the algorithm converges quickly. Figure 1 shows the functional dependencies, and Figure 2 shows the sequence of steps of the algorithm, which is detailed in the sequel, for the illustrative case of Poisson response and logarithmic link function. For other models, the particular expressions should be substituted accordingly, but the steps of the algorithm remain the same.

We start with the description of the iterative loop; then, we show how the initial estimate $\mathbf{b}^{(1)}$ for the parameter vector can be obtained, for use in the first iteration.

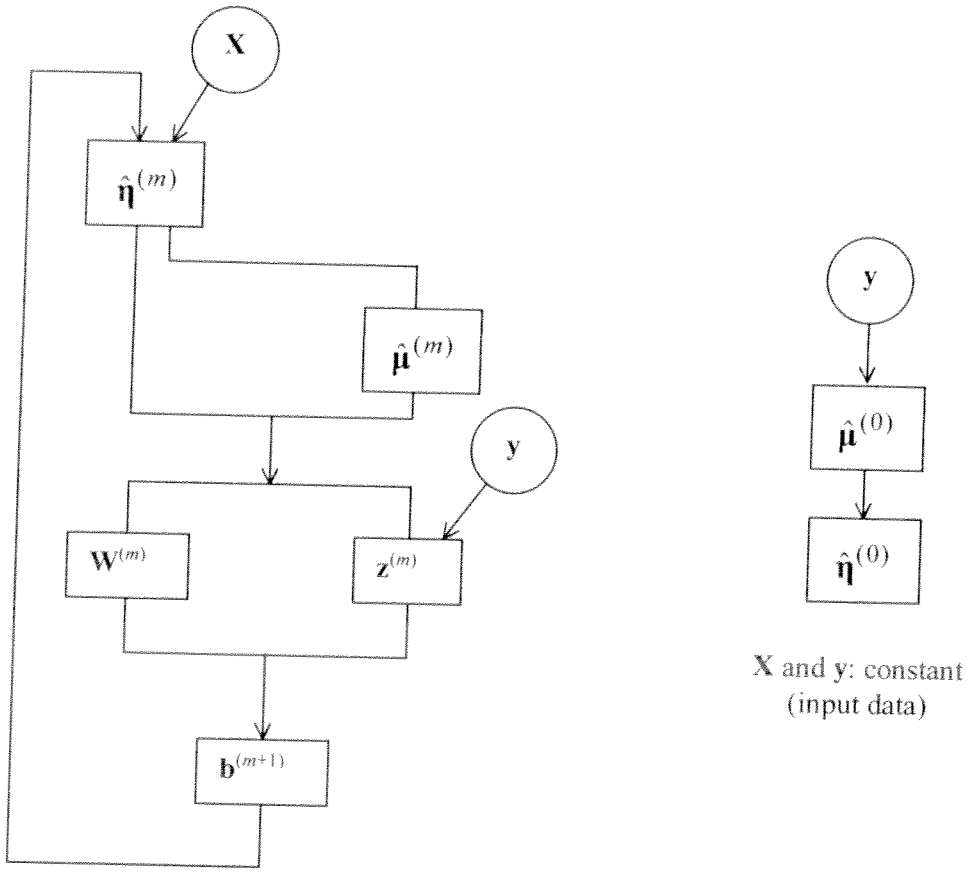

Figure 1 – Functional dependencies

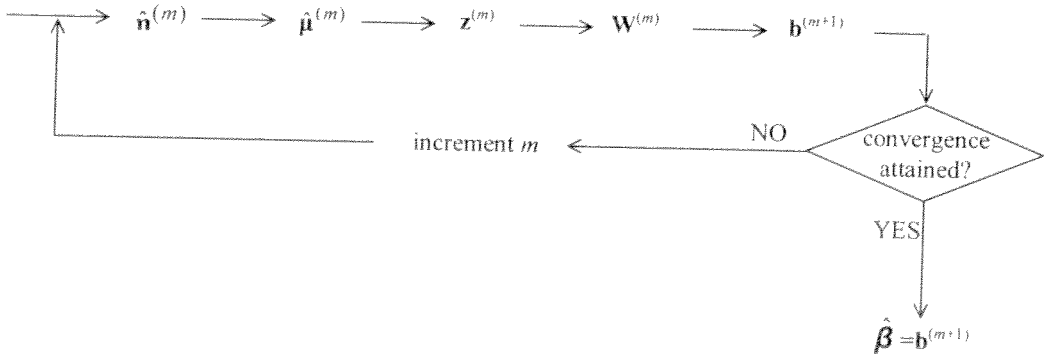

Figure 2 – ISRL algorithm

Iterative loop

Let m indicate the iteration; the m-th iteration makes use of the estimate $\mathbf{b}^{(m)}$ of the parameter vector, derives from it the estimates of η, μ, $\mathbf{z}$ and $\mathbf{W}$, and ends by determining the updated estimate $\mathbf{b}^{(m+1)}$ to be used in the next iteration.

Step 1: Compute the vector $\hat{\eta}^{(m)}$:

$$\hat{\eta}^{(m)} = \begin{bmatrix} \hat{\eta}_1^{(m)} \\ \hat{\eta}_2^{(m)} \\ \vdots \\ \hat{\eta}_k^{(m)} \end{bmatrix} = \mathbf{X}\mathbf{b}^{(m)} \tag{16}$$

Step 2: Compute the vector:

$$\hat{\mu}^{(m)} = \begin{bmatrix} \hat{\mu}_1^{(m)} \\ \hat{\mu}_2^{(m)} \\ \vdots \\ \hat{\mu}_k^{(m)} \end{bmatrix} \tag{17}$$

where each $\hat{\mu}_i$ is obtained inverting the link function. For the case of the link function in Eq. (6), this gives:

$$\hat{\mu}_i^{(m)} = \exp\left(\hat{\eta}_i^{(m)}\right) \qquad i = 1, 2, \cdots, n \tag{18}$$

Step 3: Compute the vector **z**. In the case of the link function in Eq. (6), Eq. (15) for the elements of **z** becomes:

$$z_i^{(m)} = \hat{\eta}_i^{(m)} + \frac{\left(y_i - \hat{\mu}_i^{(m)}\right)}{\hat{\mu}_i^{(m)}} \qquad i = 1, 2, \cdots, n \tag{19}$$

Step 4: Compute the elements of the diagonal matrix **W**, given by (13), which, with Poisson response and link function given by (6), become:

$$w_{ii}^{(m)} = \hat{\mu}_i^{(m)} \qquad i = 1, 2, \cdots, n \tag{20}$$

Step 5: Update the estimate vector **b**:

$$\mathbf{b}^{(m+1)} = \left(\mathbf{X}'\mathbf{W}^{(m)}\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{W}^{(m)}\mathbf{z}^{(m)} \tag{9, repeated}$$

Step 6: Test the convergence of the algorithm, by verifying if some appropriate (user-defined) measure of distance between $\mathbf{b}^{(m+1)}$ and $\mathbf{b}^{(m)}$ is smaller than a specified tolerance value. If the convergence has occurred, END the algorithm. If not, increment *m* and go back to **Step 1**, for the next iteration.

Obtaining the initial estimate $\mathbf{b}^{(1)}$ for the parameter vector

There is no unique and general method for obtaining the initial estimate $\mathbf{b}^{(1)}$. One of the most commonly used procedures – which will be adopted here – is to estimate $\mathbf{b}^{(1)}$ from an initial estimate $\hat{\mu}^{(0)}$ for the mean vector μ. A reasonable initial estimate is $\hat{\mu}^{(0)}$=y. Note

that in the Iterative Loop the obtention of $\hat{\mu}^{(0)}$ constitutes Step 2; so, the initial estimate $b^{(1)}$ can be obtained from the initial estimate $\hat{\mu}^{(0)}$, proceeding through Steps 3, 4 and 5. There is, nevertheless, a difference with respect to the iterations of the Iterative Loop: it will be necessary to calculate the values $\hat{\eta}_i^{(0)} = g(\hat{\mu}_i^{(0)})$, because Step 3 requires an estimate of η. To sum all up, the sequence becomes:

a) Set $\hat{\mu}_i^{(0)} = y_i \qquad i = 1, 2, \cdots, n$ (21)

b) Set $\hat{\eta}_i^{(0)} = g\left(\hat{\mu}_i^{(0)}\right) \qquad i = 1, 2, \cdots, n$ (22)

which, in our case, due to (6) becomes:

$$\hat{\eta}_i^{(0)} = \ln \hat{\mu}_i^{(0)} \qquad i = 1, 2, \cdots, n \tag{23}$$

c) Execute Step 3 of the Iterative Loop. Note that, in the initialization, because of (21), Eq. (19) in Step 3 becomes simply:

$$z_i^{(0)} = \hat{\eta}_i^{(0)} \qquad i = 1, 2, \cdots, n \tag{24}$$

which in turn, in our model, because of (23) and (21), becomes:

$$z_i^{(0)} = \ln y_i \qquad i = 1, 2, \cdots, n \tag{25}$$

d) Execute Steps 4 and 5 of the Iterative Loop, obtaining $\mathbf{b}^{(1)}$.

Spreadsheet Implementation of the Algorithm

Let us illustrate the spreadsheet implementation of the algorithm by the analysis of a simulated experiment with an unreplicated 2^3 factorial design. It will be assumed that the reader knows how to invert, transpose and multiply matrices using the spreadsheet.

Underlying model and simulated data

The data were randomly generated according to the following multiplicative model for the mean:

$$\mu = 10(1.5)^{x_1}(0.7)^{x_2}(1.4)^{x_2 x_3} \tag{26}$$

where: 10 is the base mean μ_0; the factors A, B and C are represented by the coded variables x_1, x_2 and x_3, respectively; and 1.5, 0.7 and 1.4, are the effects of A, B and BC respectively. The other effects are not significant and therefore set equal to 1.

The model given by Eq. (26) is equivalent to:

$$\mu = \exp(2.302 + 0.405x_1 - 0.357x_2 + 0.336x_2x_3) \tag{27}$$

Table 1 presents the design along with the theoretical means μ_i calculated by (26) and the responses y_i randomly generated according to a Poisson(μ_i) distribution.

Table 1 – 2^4 Factorial Design with the Theoretical Mean and the Generated Response.

Run	x_1	x_2	x_3	μ_i	y_i
1	-1	-1	-1	13.3	14
2	1	-1	-1	30.0	29
3	-1	1	-1	3.3	4
4	1	1	-1	7.5	5
5	-1	-1	1	6.8	7
6	1	-1	1	15.3	16
7	-1	1	1	6.5	4
8	1	1	1	14.7	17

Data entry and initial estimate $\mathbf{b}^{(1)}$

The input data are the design matrix **X** (where the factor levels are represented by -1 and +1) and the response vector **y**, as in Figure 3. Note that the first column of the matrix **X** is filled with 1's, corresponding, by definition, to x_{i0}=1, for i=1, 2, ..., n. For the initialization (as seen in the subsection *"Obtaining the initial estimate* $\mathbf{b}^{(1)}$ *for the parameter vector"*), one should set $\hat{\mu}_i^{(0)} = y_i$, and therefore In $\hat{\eta}_i^{(0)}$ = In $\hat{\mu}_i^{(0)}$ = In y_i, and then proceed through steps 3, 4 and 5 as follows. First, Eq. (19) in Step 3 becomes simply $z_i^{(0)} = \hat{\eta}_i^{(0)}$ (Eq. 13), so the column with elements $z_i^{(0)}$ can be obtained simply by setting $z_i^{(0)}$ = In y_i. Step 4 (obtention of the matrix **W**) is performed in 3 stages: first, by generating the elements w_{ii} in a separate column; next, by filling an nxn matrix with zeros; and finally, by copying the elements w_{ii} (one by one) to the respective cell of the diagonal of the matrix, overwriting the zeros. To generate the column of elements w_{ii}, remember that each w_{ii} is the variance of y_i, which, in the case of the Poisson distribution, equals (see Eq. 10) the mean μ_i – whose initial estimate in turn, as seen above and in Eq. (21), equals the response value y_i itself. Therefore, in the particular case of the Poisson model, the column of elements $w_{ii}^{(0)}$ is identical to the column that contains the vector **y**.

Step 5 is to compute $\mathbf{b}^{(1)}$ by Eq. (9). This involves the following sequence of operations: transpose the matrix **X**, obtaining **X**′; next multiply **X**′ by $\mathbf{W}^{(0)}$ to obtain the matrix $\mathbf{X}'\mathbf{W}^{(0)}$, which should be in turn multiplied by **X** to obtain the matrix $\mathbf{X}'\mathbf{W}^{(0)}\mathbf{X}$; now invert this matrix to obtain $(\mathbf{X}'\mathbf{W}^{(0)}\mathbf{X})^{-1}$; then multiply $\mathbf{X}'\mathbf{W}^{(0)}$ by the vector $\mathbf{z}^{(0)}$, obtaining the column vector $\mathbf{X}'\mathbf{W}^{(0)}\mathbf{z}^{(0)}$; and finally multiply the matrix $(\mathbf{X}'\mathbf{W}^{(0)}\mathbf{X})^{-1}$ by the column vector $\mathbf{X}'\mathbf{W}^{(0)}\mathbf{z}^{(0)}$, which gives $\mathbf{b}^{(1)}$.

Figure 4 shows all intermediate matrices, in the order in which they are calculated (from left to right and from top to bottom), as well as the resulting vector $\mathbf{b}^{(1)}$.

X							y	$z^{(0)}$ $z_i = \ln y_i$	w_{ii}	$W^{(0)}$							
	x_1	x_2	$x_1 x_2$	x_3	$x_1 x_3$	$x_2 x_3$											
1	-1	-1	1	-1	1	1	14	2.64	14	14	0	0	0	0	0	0	0
1	1	-1	-1	-1	-1	1	29	3.37	29	0	29	0	0	0	0	0	0
1	-1	1	-1	-1	1	-1	4	1.39	4	0	0	4	0	0	0	0	0
1	1	1	1	-1	-1	-1	5	1.61	5	0	0	0	5	0	0	0	0
1	-1	-1	1	1	-1	-1	7	1.95	7	0	0	0	0	7	0	0	0
1	1	-1	-1	1	1	-1	16	2.77	16	0	0	0	0	0	16	0	0
1	-1	1	-1	1	-1	1	4	1.39	4	0	0	0	0	0	0	4	0
1	1	1	1	1	1	1	17	2.83	17	0	0	0	0	0	0	0	17

Figure 3 – Input data and calculation of the matrix $W^{(0)}$

X'								$X'W^{(0)}$								$X'W^{(0)}X$						
1	1	1	1	1	1	1	1	14	29	4	5	7	16	4	17	96	38	-36	-10	-8	6	32
-1	1	-1	1	-1	1	-1	1	-14	29	-4	5	-7	16	-4	17	38	96	-10	-36	6	-8	18
-1	-1	1	1	-1	-1	1	1	-14	-29	4	5	-7	-16	4	17	-36	-10	96	38	32	18	-8
1	-1	-1	1	1	-1	-1	1	14	-29	-4	5	7	-16	-4	17	-10	-36	38	96	18	32	6
-1	-1	-1	-1	1	1	1	1	-14	-29	-4	-5	7	16	4	17	-8	6	32	18	96	38	-36
1	-1	1	-1	-1	1	-1	1	14	-29	4	-5	-7	16	-4	17	6	-8	18	32	38	96	-10
1	1	-1	-1	-1	-1	1	1	14	29	-4	-5	-7	-16	4	17	32	18	-8	6	-36	-10	96

$(X'W^{(0)}X)^{-1}$							$X'W^{(0)}z^{(0)}$	$b^{(1)}$	
0.0166	-0.0059	0.0067	-0.0020	-0.0008	-0.0022	-0.0043	2.255	260	Intercept
-0.0059	0.0159	-0.0026	0.0068	-0.0033	0.0009	-0.0028	0.435	137	x1
0.0067	-0.0026	0.0162	-0.0057	-0.0041	-0.0003	-0.0016	-0.414	-125	x2
-0.0020	0.0068	-0.0057	0.0167	-0.0016	-0.0035	-0.0031	0.020	-46	x1 x2
-0.0008	-0.0033	-0.0041	-0.0016	0.0165	-0.0048	0.0063	0.008	-36	x3
-0.0022	0.0009	-0.0003	-0.0035	-0.0048	0.0138	0.0004	0.106	10	x1 x3
-0.0043	-0.0028	-0.0016	-0.0031	0.0063	0.0004	0.0148	0.361	117	x2 x3

Figure 4 – Completion of the initialization with the intermediate matrices and the initial estimate $b^{(1)}$

From this point on, the iterative process begins. As will be seen, it is sufficient to set up the spreadsheet for the first iteration; a neat trick enables performing subsequent iterations by an elementary action (this renders the time to perform one iteration human-bound, but in practice this does not constitute a drawback, since the algorithm converges very quickly). Only the first iteration is different, as it involves building the spreadsheet for the Iterative Loop. It will be described now, step by step.

First iteration

For the sake of clarity, we will construct a separate spreadsheet for the Iterative Loop, as shown in Figure 5. We start by copying the column vector $\mathbf{b}^{(1)}$ and the matrix $\mathbf{X}$, placing them side by side, as shown in Block 1 of Figure 5. From this point on, we follow the steps of the Iterative Loop. Since this spreadsheet is set for all iterations, $\mathbf{b}^{(1)}$ in Figure 5 receives the heading $\mathbf{b}(m)$. For simplicity, in the spreadsheet, η, μ, $\mathbf{W}$ and $\mathbf{z}$ did not receive any index (there is no ambiguity, since the index is always m), the "hats" over η and μ were supressed, and the matrix $(\mathbf{X}'\mathbf{W}^{(0)}\mathbf{X})^{-1}$ is noted $(\mathbf{X}'\mathbf{W}\mathbf{X})$-1. Only the vector $\mathbf{b}$ remains indexed in order to distinguish between $\mathbf{b}^{(m+1)}$ and $\mathbf{b}^{(m)}$.

Step 1: *Computing the vector* $\hat{\boldsymbol{\eta}}^{(m)} = \mathbf{X}\mathbf{b}^{(m)}$

Multiply the matrix $\mathbf{X}$ by the column vector $\mathbf{b}^{(m)}$ inserting the result in a column vector "η" (column I, Block 1, in Figure 5).

Step 2: *Computing the vector of means* $\hat{\mu}^{(m)}$

Calculate the elements of the vector $\hat{\mu}^{(m)}$ (column J, Block 1, in Figure 5), from the elements of column $\hat{\eta}^{(m)}$, by Eq. (18). Each element of $\hat{\mu}^{(m)}$ is the exponential of the corresponding element of the vector $\hat{\eta}^{(m)}$ (column I).

Step 3: *Computing the vector* $\mathbf{z}^{(m)}$.

Use Eq. (19), with the elements of the vectors η (column I), μ (column J) and $\mathbf{y}$ (column H) as arguments. Enter the vector $\mathbf{z}$ in column K.

Step 4: *Computing the matrix* $\mathbf{W}$.

Matrix $\mathbf{W}$ is constructed as before, in the initialization: first a column of values w_{ii} is generated (column M, Block 1); next an nxn matrix is filled with zeros; and finally the w_{ii} values are copied into the cells of the matrix diagonal. The w_{ii} values are obtained from Eq. (20): it is sufficient to set their elements equal to the corresponding elements of column μ. Figure 5 shows the matrix $\mathbf{W}$ in the columns I to P of Block 2.

Step 5: *Updating the estimate* $\mathbf{b}$ *of the parameter vector* β.

As seen, this is done according to Eq. (9) and involves a sequence of operations. First, construct the matrix $\mathbf{X}'$ and the product matrices $\mathbf{X}'\mathbf{W}$ and $\mathbf{X}'\mathbf{W}\mathbf{X}$. Next, calculate the inverse matrix $(\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}$ and the product matrix $\mathbf{X}'\mathbf{W}\mathbf{z}$. Finally, multiply $(\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}$ and $\mathbf{X}'\mathbf{W}\mathbf{z}$, obtaining the column vector $\mathbf{b}^{(2)}$ (under the heading $\mathbf{b}(m+1)$)..

Step 6: *Testing the convergence of the solution.*

Convergence is considered to have occurred when $\mathbf{b}^{(m)}$ and $\mathbf{b}^{(m+1)}$ are close enough, according to some appropriate, user-defined, measure of distance. We may enter the formula of the measure of distance chosen into any cell of the spreadsheet for automatic calculation. Alternatively, we can check by mere visual inspection whether the corresponding elements of $\mathbf{b}^{(m)}$ and $\mathbf{b}^{(m+1)}$ are identical up to a predefined number of decimal places. (This

is equivalent in formal terms to using the greatest absolute difference $\left|\hat{\beta}_i^{m+1} - \hat{\beta}_i^{m}\right|$, i=1, 2, ...n, as measure of distance between $\mathbf{b}^{(m)}$ and $\mathbf{b}^{(m+1)}$).

Using this criterion, we observe that the estimates in $\mathbf{b}^{(m)}$ and $\mathbf{b}^{(m+1)}$ still differ in the second decimal place; we therefore return to **Step 1** for the second iteration.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1																
2	**Block 1**															
3				**X**												
4		x1	x2	x1 x2	x3	x1 x3	x2 x3	y	η	μ	z		*wi*		**b(m)**	
5	1	-1	-1	1	-1	1	1	14	2,7	15,1	2,6		15,1		2,255	
6	1	1	-1	-1	-1	-1	1	29	3,3	28,0	3,4		28,0		0,435	
7	1	-1	1	-1	-1	1	-1	4	1,1	3,1	1,4		3,1		-0,414	
8	1	1	1	1	-1	-1	-1	5	1,8	6,2	1,6		6,2		0,020	
9	1	-1	-1	1	1	-1	-1	7	1,8	6,0	2,0		6,0		0,008	
10	1	1	-1	-1	1	1	-1	16	2,8	17,1	2,8		17,1		0,106	
11	1	-1	1	-1	1	-1	1	4	1,6	5,2	1,4		5,2		0,361	
12	1	1	1	1	1	1	1	17	2,8	16,0	2,8		16,0			
13																
14																
15	**Block 2**															
16				**X'**								**W**				
17	1	1	1	1	1	1	1	1	15,1	0,0	0,0	0,0	0,0	0,0	0,0	0,0
18	-1	1	-1	1	-1	1	-1	1	0,0	28,0	0,0	0,0	0,0	0,0	0,0	0,0
19	-1	-1	1	1	-1	-1	1	1	0,0	0,0	3,1	0,0	0,0	0,0	0,0	0,0
20	1	-1	-1	1	1	-1	-1	1	0,0	0,0	0,0	6,2	0,0	0,0	0,0	0,0
21	-1	-1	-1	-1	1	1	1	1	0,0	0,0	0,0	0,0	6,0	0,0	0,0	0,0
22	1	-1	1	-1	-1	1	-1	1	0,0	0,0	0,0	0,0	0,0	17,1	0,0	0,0
23	1	1	-1	-1	-1	-1	1	1	0,0	0,0	0,0	0,0	0,0	0,0	5,2	0,0
24									0,0	0,0	0,0	0,0	0,0	0,0	0,0	16,0
25																
26																
27	**Block 3**															
28				**X'W**								**X'WX**				
29	15	28	3	6	6	17	5	16	97	38	-36	-10	-8	6	32	
30	-15	28	-3	6	-6	17	-5	16	38	97	-10	-36	6	-8	9	
31	-15	-28	3	6	-6	-17	5	16	-36	-10	97	38	32	9	-8	
32	15	-28	-3	6	6	-17	-5	16	-10	-36	38	97	9	32	6	
33	-15	-28	-3	-6	6	17	5	16	-8	6	32	9	97	38	-36	
34	15	-28	3	-6	-6	17	-5	16	6	-8	9	32	38	97	-10	
35	15	28	-3	-6	-6	-17	5	16	32	9	-8	6	-36	-10	97	
36																
37																
38	**Block 4**															
39				**(X'WX)-1**					**X'Wz**	**b(m+1)**						
40	0,017	-0,006	0,007	-0,002	-0,001	-0,001	-0,005		260	2,242	Intere					
41	-0,006	0,015	-0,003	0,006	-0,001	0,000	-0,001		134	0,442	x1					
42	0,007	-0,003	0,017	-0,007	-0,006	0,002	-0,002		-126	-0,424	x2					
43	-0,002	0,006	-0,007	0,017	0,001	-0,005	-0,001		-46	0,024	x1 x2					
44	-0,001	-0,001	-0,006	0,001	0,016	-0,006	0,006		-37	0,010	x3					
45	-0,001	0,000	0,002	-0,005	-0,006	0,014	0,000		14	0,108	x1 x3					
46	-0,005	-0,001	-0,002	-0,001	0,006	0,000	0,014		113	0,366	x2 x3					
47																

Figure 5 – Results of Iteration 1.

Second and subsequent iterations

Figure 5 shows all the first iteration data. The spreadsheet is now ready for each subsequent iteration, which can then be executed using an elementary action: transfer the new estimate $\mathbf{b}^{(2)}$ (column B in Block 4, denoted $\mathbf{b}(m+1)$ in the figure) to the cells containing the previous estimate (column 0 in Block 1), using the commands *"Copy"* and *"Paste special... Values"*. This will generate another complete iteration, transforming the spreadsheet in Figure 5 into the spreadsheet in Figure 6 (without the columns StdErr, ChiSq and *P*-value in Block 4), which shows all the second iteration data, including the new estimate $\mathbf{b}^{(3)}$ (denoted $\mathbf{b}(m+1)$ in Figure 6).

From this point on, every subsequent iteration can be generated by repeating this *"Copy/Paste special... Values"* command. In Microsoft Excel®, this can be done simply by pressing the "F4" key, or by clicking the mouse on the "paste" button (with the clipboard icon) on the standard toolbar.

Comparing the current estimate of the parameters in Figure 6 (column K, Block 4) with the estimate from the previous iteration (column 0 in Block 1) we may consider the convergence to have been achieved.

Block 1

X										
	x1	x2	x1 x2	x3	x1 x3	x2 x3	y	η	μ	z
1	-1	-1	1	-1	1	1	14	2.7	15.0	2.6
1	1	-1	-1	-1	-1	1	29	3.3	28.0	3.4
1	-1	1	-1	-1	1	-1	4	1.1	3.0	1.4
1	1	1	1	-1	-1	-1	5	1.8	6.0	1.6
1	-1	-1	1	1	-1	-1	7	1.8	6.0	2.0
1	1	-1	-1	1	1	-1	16	2.8	17.0	2.8
1	-1	1	-1	1	-1	1	4	1.6	5.1	1.4
1	1	1	1	1	1	1	17	2.8	16.0	2.8

Block 2

X′										
1	1	1	1	1	1	1	1	15.0	0.0	0.0
-1	1	-1	1	-1	1	-1	1	0.0	28.0	0.0
-1	-1	1	1	-1	-1	1	1	0.0	0.0	3.0
1	-1	-1	1	1	-1	-1	1	0.0	0.0	0.0
-1	-1	-1	-1	1	1	1	1	0.0	0.0	0.0
1	-1	1	-1	-1	1	-1	1	0.0	0.0	0.0
1	1	-1	-1	-1	-1	1	1	0.0	0.0	0.0
								0.0	0.0	0.0

Block 3

X′W										
15	28	3	6	6	17	5	16	96	38	-36
-15	28	-3	6	-6	17	-5	16	38	96	-10
-15	-28	3	6	-6	-17	5	16	-36	-10	96
15	-28	-3	6	6	-17	-5	16	-10	-36	38

Figure 6 – Results of Iteration 2.

Model checking and refinement

The next step of the analysis is to test the effects (parameters) for significance, in order to retain only the significant ones, obtaining thereby a more parsimonious model. In Figure 6, $(\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}$ is the covariance matrix of the parameters. Now look at Block 4 of the same figure. The estimated standard errors of the parameters are just the square roots of the ele-

ments of the diagonal of $(\mathbf{X}'\mathbf{WX})^{-1}$. These estimates are displayed in column K. Applying Wald's test statistic:

$$\chi_1^2 = \left[\frac{\hat{\beta}_j}{\text{StdErr}(\hat{\beta}_j)}\right]^2, \tag{28}$$

the test values (ChiSq) are in column N, and the corresponding *P*-values are in column O.

The *P*-values corresponding to x_1, x_2, $x_2 x_3$ and to the intercept are all smaller than 0.0025 (the other ones are all greater than 0.36), so we conclude by the significance of the corresponding effects (A, B and BC interaction) and retain only the parameters β_0, β_1, β_2 and β_6.

The reduced model spreadsheet is set up in the same way as for the complete model. Figure 7 shows the first iteration for the reduced model.

The initial estimate vector **b** for the reduced model is then simply the vector $(\hat{\beta}_0 \quad \hat{\beta}_1 \quad \hat{\beta}_2 \quad \hat{\beta}_6)^{(3)}$, retaining only the final estimates for the parameters β_0, β_1, β_2 and β_6 obtained with the full model.

X

	x1	x2	x2 x3	y	η	μ	z	*wii*	b(m)
1	-1	-1	1	14	2.6	13.3	2.6	13.3	2.242
1	1	-1	1	29	3.5	32.3	3.4	32.3	0.442
1	-1	1	-1	4	1.0	2.7	1.5	2.7	-0.424
1	1	1	-1	5	1.9	6.6	1.6	6.6	0.366
1	-1	-1	-1	7	1.9	6.4	1.9	6.4	
1	1	-1	-1	16	2.7	15.5	2.8	15.5	
1	-1	1	1	4	1.7	5.7	1.4	5.7	
1	1	1	1	17	2.6	13.8	2.9	13.8	

X′

1	1	1	1	1	1	1	1
-1	1	-1	1	-1	1	-1	1
-1	-1	1	1	-1	-1	1	1
1	1	-1	-1	-1	-1	1	1

W

13.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	32.3	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	2.7	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	6.6	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	6.4	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	15.5	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	5.7	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	13.8

X′W

13	32	3	7	6	16	6	14
-13	32	-3	7	-6	16	-6	14
-13	-32	3	7	-6	-16	6	14
13	32	-3	-7	-6	-16	6	14

X′WX

96	40	-39	34
40	96	-16	14
-39	-16	96	-14
34	14	-14	96

(X′WX)-1

0.016	-0.005	0.005	-0.004
-0.005	0.013	0.000	0.000
0.005	0.000	0.012	0.000
-0.004	0.000	0.000	0.012

X′Wz

262
142
-137
121

b(m+1)

2.266	Intercept
0.419	x1
-0.394	x2
0.347	x2 x3

Figure 7 – Results of Iteration 1 for the reduced model.

Comparing the updated estimate of the parameters (cells H30 to H33) with the estimate from the previous iteration (cells L4 to L7), we see that convergence has not yet occurred.

X

	x1	x2	x2 x3	y	μ	η	z	*wii*
1	-1	-1	1	14	13.3	2.6	2.6	13.3
1	1	-1	1	29	30.7	3.4	3.4	30.7
1	-1	1	-1	4	3.0	1.1	1.4	3.0
1	1	1	-1	5	7.0	1.9	1.7	7.0
1	-1	-1	-1	7	6.7	1.9	1.9	6.7
1	1	-1	-1	16	15.4	2.7	2.8	15.4
1	-1	1	1	4	6.0	1.8	1.5	6.0
1	1	1	1	17	14.0	2.6	2.9	14.0

b(m)

2.266
0.419
-0.394
0.347

X'

1	1	1	1	1	1	1	1
-1	1	-1	1	-1	1	-1	1
-1	-1	1	1	-1	-1	1	1
1	1	-1	-1	-1	-1	1	1

W

13.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	30.7	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	3.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	7.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	6.7	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	15.4	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	6.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	14.0

X'W

13	31	3	7	7	15	6	14
-13	31	-3	7	-7	15	-6	14
-13	-31	3	7	-7	-15	6	14
13	31	-3	-7	-7	-15	6	14

X'WX

96	38	-36	32
38	96	-14	13
-36	-14	96	-12
32	13	-12	96

(X'WX)-1

0.015	-0.005	0.005	-0.004
-0.005	0.012	0.000	0.000
0.005	0.000	0.012	0.000
-0.004	0.000	0.000	0.012

X'Wz

259
136
-130
116

b(m+1)

2.265	Interc.	0.124	334.4	0.0000
0.419	x1	0.111	14.205	0.0002
-0.394	x2	0.110	12.834	0.0003
0.347	x2 x3	0.108	10.258	0.0014

Figure 8 shows the second iteration for the reduced model.

The updated estimate of the parameters (cells H30 to H33) now differs from the previous one (cells L4 to L7) only by 0.001 in the first element, which can be considered as indicating convergence.

The low *P*-values associated (cells L30 to L33) confirm the adequacy of the model. The response model obtained is then:

$$\hat{\mu} = exp(2.265+0.419x_1-0.394x_2+0.347x_2x_3) \quad (29)$$

For the small number of data points used, these coefficients are very close to those of the theoretical model:

$$\hat{\mu} = exp(2.302+0.405x_1-0.357x_2+0.336x_2x_3) \quad (30)$$

The precision of the results can also be assessed in terms of the difference between the estimated and the theoretical means for each run. These are given in Table 2.

Table 2 – Theoretical and estimated means.

Run	Theoretical mean	Estimated mean	Difference (%)
1	13,33	13,29	0,30
2	30,00	30,71	-2,31
3	3,33	3,02	10,26
4	7,50	6,98	7,45
5	6,80	6,65	2,26
6	15,31	15,35	-0,26
7	6,53	6,04	8,11
8	14,70	13,96	5,30

Numerical precision issues

The numerical precision of the elements of inverse matrices is very sensitive to the particular inversion algorithm employed and also sensitive to the internal precision (number of significant digits) used by the software. Spreadsheet software is not, in general, as precise in performing matrix inversion as are statistical packages of well-proven efficiency such as (for instance) SAS, so some caution is required. Nevertheless, the use of spreadsheet software for solving GLMs is proposed here for occasional users, for small problems, or even for teaching and training purposes. If this is the case, the small size of the matrices involved ensures good precision of the results (unless the matrix to be inverted has a very particular structure which is not likely to arise in GLMs). The user can always check the precision of $(\mathbf{XWX})^{-1}$ simply by multiplying it by $\mathbf{XWX}$ and comparing the result with the identity matrix, if desired. However, we tested the procedure by analyzing the data sets analyzed by Hamada and Nelder (1997) and by Myers and Montgomery (1997), who used GENSTAT and SAS. Our final results were identical to theirs up to the last digit (the number of significant digits provided in the references ranged from three to five); so the precision of the results can be considered satisfactory for the purposes of statistical analysis in view of the existence of experimental errors.

Conclusions

We have shown how analysis of experiments using Generalized Linear Models can be performed with the help of spreadsheet software. Once the spreadsheet is set up for the first iteration, every additional iteration can be performed with an elementary operation; in Microsoft Excel®, a single mouse click on the "paste" button, or a stroke on the "F4" key, is sufficient. The number of iterations required for convergence is very small.

The use of the spreadsheet software for GLM implementation was illustrated for the particular case of a Poisson model for the response, with logarithmic link function, but the procedure is general. For other models of the response and link function, the particular expressions for calculating the elements of the vector $\hat{\mu}$ of estimated means by the inverse of the link function (Eq. 8), the elements of the diagonal matrix **W** (Eq. 10) and of the vector **z** of adjusted variables (Eq. 9) will change accordingly; namely, the expressions to be used will be the inverse of the link function and expressions derived from the generic Eqs. (13) and (15) in accordance with the model adopted.

Spreadsheet solution of GLMs may fit the needs of occasional users of GLMs, for small problems, and has the advantage of being accessible to everyone. It is also interesting for teaching and training purposes, in illustrating the use of GLMs in data analysis in general, and because it exposes the mechanics of the solution algorithm.

Acknowledgements

The second author was partly supported by the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Brazil.

References

Bisgaard, S. and Fuller, H.T. (1994-95) Analysis of Factorial Experiments with Defects or Defectives as the Response, Quality Engineering, Vol. 7, pp. 429–443.

Box, G. and Fung, C.A. (1995) Quality Quandaries, Quality Engineering, Vol. 7, pp. 625–638.

Dobson, A.J. (1990) An Introduction to Generalized Linear Models, Chapman and Hall/CRC, New York.

Hamada, M. and Nelder, J.A. (1997) Generalized Linear Models for Quality Improvement Experiments, Journal of Quality Technology, Vol. 29 No 3, pp. 292–304.

Lewis, S.L., Montgomery, D.C. and Myers, R.H. (2001) Examples of Designed Experiments with Nonnormal Responses, Journal of Quality Technology, Vol. 33 No 3, pp. 265–278.

McCullough, P. and Nelder, J.A. (1989) Generalized Linear Models, 2nd Edition, Chapman and Hall, New York.

Myers, R.H. and Montgomery, D.C. (1997) A Tutorial on Generalized Linear Models, Journal of Quality Technology, Vol. 29 No 3, pp. 274–291.

Biography

Antonio Fernando de Castro Vieira is an Assistant Professor in the Department of Industrial Engineering at PUC – Pontifícia Universidade Católica, Rio de Janeiro, Brazil. He received his Mechanical Engineering degree from the Federal University of Pernambuco

(UFPE), Brazil, and his M.Sc. in Industrial Engineering from PUC, Rio de Janeiro. His main research interests are in statistical quality control and in design and analysis of experiments for process optimization.

Eugenio Kahn Epprecht is an Assistant Professor in the Department of Industrial Engineering at PUC – Pontifícia Universidade Católica, Rio de Janeiro, Brazil. He received his Eng. and M.Sc. degrees from PUC, Rio de Janeiro, and his Ph.D. from the Facultés Universitaires Notre-Dame de la Paix, Belgium. His current research interests include statistical quality control and industrial applications of statistics and operations research. He has published papers in *IIE Transactions*, *International Journal of Production Research*, *International Journal of Production Economics* and *Quality Engineering*, and in the Brazilian journals *Gestão & Produção*, *Pesquisa Operacional* and *Pesquisa Naval*. He is a member of the *American Society for Quality* (ASQ), *Associação Brasileira de Engenharia de Produção* (ABEPRO) and *Sociedade Brasileira de Pesquisa Operacional* (SBPO).