ORIGIN ≡ 0

GLM Test of Linear Fit

W. Stein

The General Linear Models (GLM) Test using comparison between "full" and "reduced" statistical models allows one to formally argue whether a linear fit of the data is sufficient to describe the distribution of X,Y points in a dataset, or whether a more complex non-linear model (or transformed linear model) is required instead. This example is "creepy" because on first sight the "full" model doesn't *look* as constrained as the "reduced" model. However, one can tell that the "full" model is in fact the more constrained by calculating the Sum of Squares Error for Full Model SSEF, and seeing that it is less than Sum of Squares Error for Reduced model SSER. One can also compare the total number of parameters estimated for each model. A Full Model always has more than the corresponing Reduced Model. For the Full Model here, an expected level in dependent variable Y for each "bin" of independent variable X involves estimation of a separate parameter (mean $Ybar_j$ for each bin X_i - estimating μ_j for each j). Thus, for the Full Model, a total of max(i)=c parameters are estimated. The Reduced Model, by contrast, is fit using a single regression line, and this involves estimation of only 2 parameters (b_0 & b_1 estimating β_0 & β_1 respectively). So, this set up works when c > 2.

This example is also instructive in giving us an expanded meaning for the Null Hypothesis H_0. In many GLM Full vs Reduced model tests, H_0 involves setting one or more parameters of a linear full model to zero. The two models neatly internest with "reduced" models comprising a subset of the "full" model. However, in this case Full versus Reduced Models have different forms. This points the generality and power of the GLM approach. Worked example drawn from Kuter et al. (KNNL) A*pplied Linear Statistical Models* 5th Edition.

Assumptions:

- Standard Linear Regression depends on specifying in advance which variable is to be considered 'dependent' and which 'independent'. This decision matters as changing roles for Y & X usually produces a different result.

- $Y_1, Y_2, Y_3, \ldots, Y_n$ (dependent variable) is a random sample ~ $N(\mu,\sigma^2)$.

- $X_1, X_2, X_3, \ldots, X_n$ (independent variable) with each value of X_i matched to Y_i

Within this setup, two models for the relationship between X and Y variables are explicitly compared:

Full Model:

$Y_{ij} = \mu_j + \varepsilon_{ij}$ **where: μ_j are parameter means at specified Bin levels of X. Bins are indicated by index variable j, with j having c levels with c > 2**

ε_{ij} are "within" errors compared to each mean μ_j ~$N(0,\sigma^2)$

Reduced Model:

$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$ **where: β_0 is the y intercept of the regression line (translation),**

β_1 is the slope of the regression line (scaling coefficient),

ε_i is the error factor in prediction of Y_i and a random variable ~$N(0,\sigma^2)$.

Example: Reading Bank Example KNNL Table 3.4:

K := READPRN("c:/2008LinearModelsData/Bank.txt")

^ The data was first sorted in Excel by ascending values of the first column (X) before importing.

K =

	0	1
0	75	28
1	75	42
2	100	112
3	100	136
4	125	160
5	125	150
6	150	152
7	175	156
8	175	124
9	200	124
10	200	104

Assigning Variables and Calculating Simple Statistics:

$X := K^{\langle 0 \rangle}$

$Y := K^{\langle 1 \rangle}$

$n := \text{length}(X)$ $\qquad n = 11$ **< total observations**

Setting up Bins for the Full Model:

$c := 6$ **< defined number of bins**

$j := 0 .. c-1$ **< range variable j for c bins**

$X = $

	0
0	75
1	75
2	100
3	100
4	125
5	125
6	150
7	175
8	175
9	200
10	200

$Y = $

	0
0	28
1	42
2	112
3	136
4	160
5	150
6	152
7	156
8	124
9	124
10	104

$XB_0 := \begin{pmatrix} X_0 \\ X_1 \end{pmatrix}$ $\quad YB_0 := \begin{pmatrix} Y_0 \\ Y_1 \end{pmatrix}$ $\quad XB_0 = \begin{pmatrix} 75 \\ 75 \end{pmatrix}$ $\quad YB_0 = \begin{pmatrix} 28 \\ 42 \end{pmatrix}$

$XB_1 := \begin{pmatrix} X_2 \\ X_3 \end{pmatrix}$ $\quad YB_1 := \begin{pmatrix} Y_2 \\ Y_3 \end{pmatrix}$ $\quad XB_1 = \begin{pmatrix} 100 \\ 100 \end{pmatrix}$ $\quad YB_1 = \begin{pmatrix} 112 \\ 136 \end{pmatrix}$

$XB_2 := \begin{pmatrix} X_4 \\ X_5 \end{pmatrix}$ $\quad YB_2 := \begin{pmatrix} Y_4 \\ Y_5 \end{pmatrix}$ $\quad XB_2 = \begin{pmatrix} 125 \\ 125 \end{pmatrix}$ $\quad YB_2 = \begin{pmatrix} 160 \\ 150 \end{pmatrix}$

$XB_3 := X_6$ $\quad YB_3 := Y_6$ $\quad XB_3 = 150$ $\quad YB_3 = 152$

$XB_4 := \begin{pmatrix} X_7 \\ X_8 \end{pmatrix}$ $\quad YB_4 := \begin{pmatrix} Y_7 \\ Y_8 \end{pmatrix}$ $\quad YB_4 = \begin{pmatrix} 156 \\ 124 \end{pmatrix}$ $\quad YB_4 = \begin{pmatrix} 156 \\ 124 \end{pmatrix}$

< Note: this was done by hand. Statistical software typically have automated ways to efficiently handle the problem of binning.

$XB_5 := \begin{pmatrix} X_9 \\ X_{10} \end{pmatrix}$ $\quad YB_5 := \begin{pmatrix} Y_9 \\ Y_{10} \end{pmatrix}$ $\quad XB_5 = \begin{pmatrix} 200 \\ 200 \end{pmatrix}$ $\quad YB_5 = \begin{pmatrix} 124 \\ 104 \end{pmatrix}$

Calculating Sum of Squares Error for the Full Model:

$Y_{bar_j} := \text{mean}\left(YB_j\right)$ $\qquad Y_{bar} = \begin{pmatrix} 35 \\ 124 \\ 155 \\ 152 \\ 140 \\ 114 \end{pmatrix}$

$e_j := YB_j - Y_{bar_j}$

$$\sum_j \left(YB_j - Y_{bar_j}\right)^2 = \begin{pmatrix} 574 \\ 574 \end{pmatrix}$$

^ summing the square differences between YB_j and Y_{bar}
Note that Mathcad had difficulty and reported a partial sum instead. So I added them together with the extra Σ here.

$$SSE_F := \sum \left[\sum_j \left(e_j\right)^2 \right]$$

$SSE_F = 1148$ **< Results confirmed p. 122.**

Performing Linear Regression for the Reduced Model:

Initial calculations:

$i := 0..n-1$ **< range variable i**

$X_{bar} := \text{mean}(X)$ $X_{bar} = 136.3636$ $Y_{bar} := \text{mean}(Y)$ $Y_{bar} = 117.0909$ **< means for X & Y**

$L_{xx} := \sum_i \left(X_i - X_{bar}\right)^2$ $L_{xx} = 21704.5455$ **< Corrected Sum of Squares for X_i**

$L_{yy} := \sum_i \left(Y_i - Y_{bar}\right)^2$ $L_{yy} = 19882.9091$ **< Corrected Sum of Squares for Y_i**

$L_{xy} := \sum_i \left(X_i - X_{bar}\right)\cdot\left(Y_i - Y_{bar}\right)$ $L_{xy} = 10563.6364$ **< Corrected Sum of Squares for Cross Product**

Regression coefficients:

$b_1 := \frac{L_{xy}}{L_{xx}}$ $b_1 = 0.4867$ **< sample estimate of slope β_1**

$b_0 := Y_{bar} - b_1 \cdot X_{bar}$ $b_0 = 50.7225$ **< sample estimate of intercept β_0**

Point Estimate of mean response (i.e., Regression line):

$Yh_i := b_0 + b_1 \cdot X_i$ **< vector of points along the regression line.**

Residuals:

$e_i := Yh_i - Y_i$ **< vector of deviations of each value Y_i from Regression line = Yh_i**

Calculating Sum of Squares Error for the Reduced Model:

$SSE_R := \sum_i \left(e_i\right)^2$ $SSE_R = 14741.5707$ **< Results confirmed p. 123**

Hypotheses:

H_0: $E(Y) = \beta_0 + \beta_1 X$, that is a Linear association between Y & X

H_1: E(Y) requires more specification than linear association such as μ_j with j up to c > 2

Note that Null Hypotheses are, in general, a formal statement of parsimony (i.e., "simplicity" of explanation). The null hypothesis says that the simpler of two alternatives is to be preferred unless the data require us to reject it. In a one population t-test of mean, for example, we ask whether an observed mean value X_{bar} is statistically indistinguishable from some specified value μ_0. We normally interpret the Null Hypothesis H_0 to say "the differences we observe between X_{bar} and μ_0 are the expected result of random behavior". However, random must always be defined in light of some model of what we expect for random, such as ~N(μ,σ). We might more accurately claim instead that H_0 says "unless compelled to do so, prefer the simpler hypothesis about difference between X_{bar} and μ_0", namely that there is nothing more to explain about the relationship between X_{bar} and $\mu_0$than ~N(μ,σ) . The Null Hypothesis for GLM above works exactly this way.

Degrees of Freedom:

$df_F := n - c \quad df_F = 5$ < **"full" model with c estimated parameters for bin means**

$df_R := n - 2 \quad df_R = 9$ < **"reduced" model with 2 parameters for regression coefficents**

GLM Test Statistic:

$$F := \frac{\frac{SSE_R - SSE_F}{df_R - df_F}}{\frac{SSE_F}{df_F}} \qquad F = 14.8014$$

< **results confirmed p. 124**

Critical Value of the Test:

$\alpha := 0.01$ < **Probability of Type I error must be explicitly set**

$CV := qF(1 - \alpha, c - 2, n - c) \quad CV = 11.3919$ < **note degrees of freedom here**

^ **CV confirmed p. 124**

Decision Rule:

IF F > CV, THEN REJECT H_0 OTHERWISE ACCEPT H_0

$F = 14.8014 \qquad CV = 11.3919$

Probability Value:

$P := 1 - pF(F, c - 2, n - c) \quad P = 0.0056$ < **results confirmed p. 124.**

Note on Bins:

Although this test is formally designed to cover X,Y data with exact replicates in X, KNNL allow that sets of nearby X's with *approximately similar* Y's can be binned also. Results with these so-called pseudoreplicates should then be considered only approximate. The only problem I can see with pseudoreplicate binning is that perhaps one should be careful not to "cherry pick" sets of X's so as to force Y_{bar}'s into a specific *a priori* pattern. Doing so necessarily biases conclusions against parsimony as described above, resulting in rejection of H_0 perhaps more than one should. On the other hand, the whole question is whether a linear model based on H_0 suffices to explain the data. If H_0 suffices despite a concerted attempt to bias against it, perhaps one can consider the case well made.

Prototype in R:

```
# READ A STRUCTURED DATA TABLE
K=read.table("c:/2008LinearModelsData/bankR.txt")
K
X=K$deposit
Y=K$newacc
B=factor(K$bin)

# FINDING N & C
n=length(X)
n
c=nlevels(B)
c
```

K

	deposit	newacc	bin
1	75	28	1
2	75	42	1
3	100	112	2
4	100	136	2
5	125	160	3
6	125	150	3
7	150	152	4
8	175	156	5
9	175	124	5
10	200	124	6
11	200	104	6

$n = 11 \qquad c = 6$

```
# ANOVA OF FITTED MODELS
anova(lm(Y~B)) #FULL MODEL
anova(lm(Y~X)) #REDUCED MODEL

# CALCULATING SUM OF SQUARES ERROR
# FOR FULL MODEL
MF=summary(lm(Y~B),digits=10)
MSEF=MF$sigma^2
dfF=MF$df[2]
SSEF=dfF*MSEF
SSEF

# CALCULATING SUM OF SQUARES ERROR
# FOR REDUCED MODEL
MR=summary(lm(Y~X),digits=10)
MSER=MR$sigma^2
dfR=MR$df[2]
SSER=dfR*MSER
SSER

# CALCULATING GLM F STATISTIC
F=((SSER-SSEF)/(dfR-dfF))/(SSEF/dfF)
F
# FINDING CRITICAL VALUE
alpha=0.01
CV=qf(1-alpha,c-2,n-c)
CV
# PROBABILITY VALUE
P=1-pf(F,c-2,n-c)
P
```

```
> anova(lm(Y~B)) #FULL MODEL
Analysis of Variance Table

Response: Y
          Df  Sum Sq Mean Sq F value   Pr(>F)
B          5 18734.9  3747.0  16.320 0.004085 **
Residuals  5  1148.0   229.6
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.'
0.1 ' ' 1
> anova(lm(Y~X)) #REDUCED MODEL
Analysis of Variance Table

Response: Y
          Df  Sum Sq Mean Sq F value Pr(>F)
X          1  5141.3  5141.3  3.1389 0.1102
Residuals  9 14741.6  1638.0
```

```
> SSEF
[1] 1148

> SSER
[1] 14741.57

> F
[1] 14.80136

> CV
[1] 11.39193

> P
[1] 0.005593812
```

```
# THE EFFICIENT WAY TO DO THIS
# SPECIFY FULL VS REDUCED MODELS:
FM=lm(Y~B)
RM=lm(Y~X)
# CALCULATE ANOVA TABLE OF COMPARISON
anova(RM,FM)
```

```
> anova(RM,FM)
Analysis of Variance Table

Model 1: Y ~ X
Model 2: Y ~ B
  Res.Df   RSS Df Sum of Sq      F   Pr(>F)
1      9 14742
2      5  1148  4     13594 14.801 0.005594 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

^ the parsimouious model is rejected...