Variable Selection:

Backwards Elimination SAS Macro

**Purpose**:

The macro performs an automated backward elimination variable selection process for PROC GENMOD which does not come with model selection options. Note that the GENMOD procedure in SAS versions prior to 9.4 does not come with model selection options.

**Introduction**:

SAS users of SAS 9.2 and prior versions may face situations where some "powerful" options are only available in certain SAS procedures but not available in others. For example, the model selection options are available in PROC REG, LOGISTIC, PHREG, etc., but not in PROC GENMOD, CATMOD, MIXED, etc. This backwards selection macro could be used with the procedures GENNMOD, CATMOD, MIXED, GLIMMIX, etc.

**Illustration:**

The following SAS statements simulate 5000 observations, which are based on an underlying Tweedie generalized linear model (GLM) that exploits its connection with the compound Poisson distribution. A natural logarithm link function is assumed for modeling the response variable (yTweedie), and there are five categorical variables (C1–C5), each of which has four numerical levels and two continuous variables (D1 and D2). By design, two of the categorical variables, C3 and C4, and one of the two continuous variables, D2, have no effect on the response. The dispersion parameter is set to 0.5, and the power parameter is set to 1.5.

%let nObs = 5000;

%let nClass = 5;

%let nLevs = 4;

%let seed = 1234;

**data** tmp1;

array c{&nClass};

keep c1-c&nClass yTweedie d1 d2;

/\* Tweedie parms \*/

phi=**0.5**;

p=**1.5**;

do i=**1** to &nObs;

do j=**1** to &nClass;

c{j} = int(ranuni(**1**)\*&nLevs);

end;

d1 = ranuni(&seed);

d2 = ranuni(&seed);

xBeta = **0.5**\*((c2<**2**) - **2**\*(c1=**1**) + **0.5**\*c&nClass + **0.05**\*d1);

mu = exp(xBeta);

/\* Poisson distributions parms \*/

lambda = mu\*\*(**2**-p)/(phi\*(**2**-p));

/\* Gamma distribution parms \*/

alpha = (**2**-p)/(p-**1**);

gamma = phi\*(p-**1**)\*(mu\*\*(p-**1**));

rpoi = ranpoi(&seed,lambda);

if rpoi=**0** then yTweedie=**0**;

else do;

yTweedie=**0**;

do j=**1** to rpoi;

yTweedie = yTweedie + rangam(&seed,alpha);

end;

yTweedie = yTweedie \* gamma;

end;

output;

end;

**run**;

The following code generates a basic explanatory data analysis for the dependent and independent variables:

/\* EDA \*/

%let var\_char = yTweedie c1 c2 c3 c4 c5 d1 d2;

%put &var\_char;

**data** var\_char;

set tmp1

(keep= &var\_char);

**run**;

**proc** **contents** data = var\_char varnum nodetails noprint

out=var\_char\_names (keep=name);

**run**;

**data** var\_char\_names;

set var\_char\_names;

j = \_n\_;

**run**;

\* Determine the number of observations;

**data** \_NULL\_;

if **0** then set var\_char\_names nobs=n;

call symputx('nrows',n);

stop;

**run**;

%put &nrows;

**%macro** ***do\_eda\_uni***;

%do obs = **1** %to &nrows;

data \_null\_;

set var\_char\_names;

if j = &obs then call symputx("var", put(name, **10.**));

run;

%if (%upcase(&var)=YTWEEDIE) or (%upcase(&var)=D1) or (%upcase(&var)=D2) %then %do;

ods graphics on;

proc means data=tmp1 fw=**12** printalltypes chartype

qmethod=os maxdec=**2**

mean

min

max

mode

range

n

nmiss

p1

p5

median

p95

p99 ;

var &var;

run;

title "histograms";

proc univariate data=tmp1 noprint;

var &var;

histogram ;

run;

ods graphics off;

%end;

%else %do;

ods graphics on;

proc freq data=tmp1

order=internal;

tables &var / scores=table plots(only)=freq;

run;

ods graphics off;

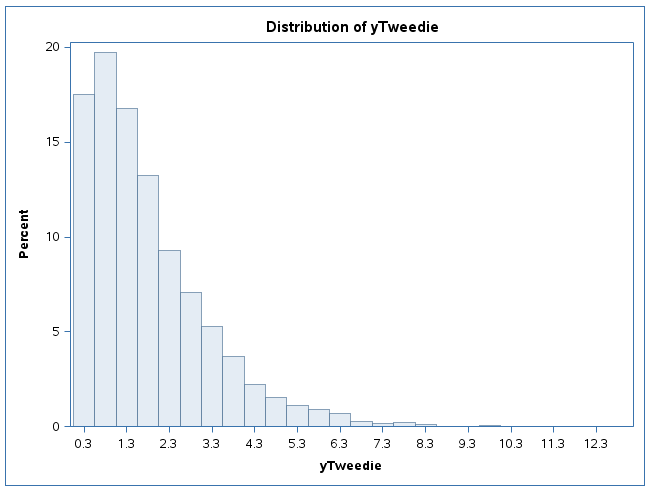
%end;

%end;

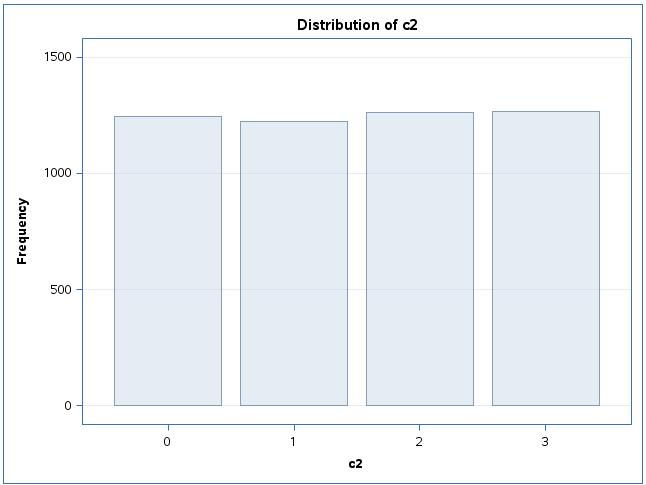
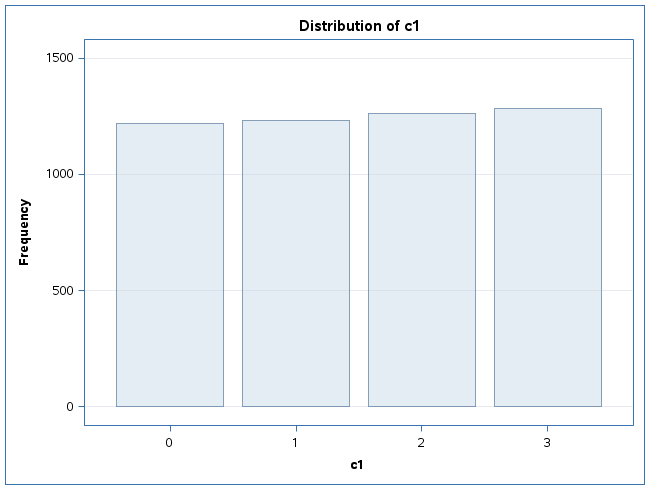
**%mend** do\_eda\_uni;

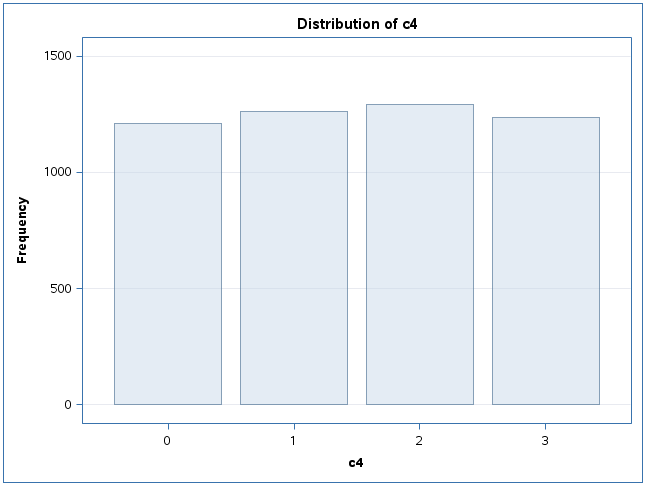
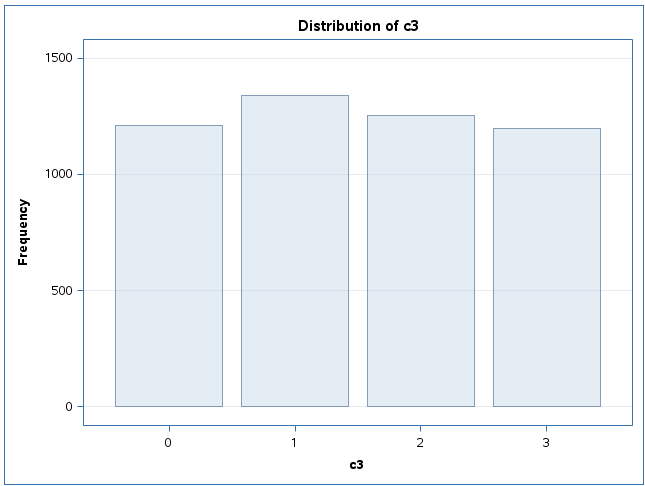
%***do\_eda\_uni***;

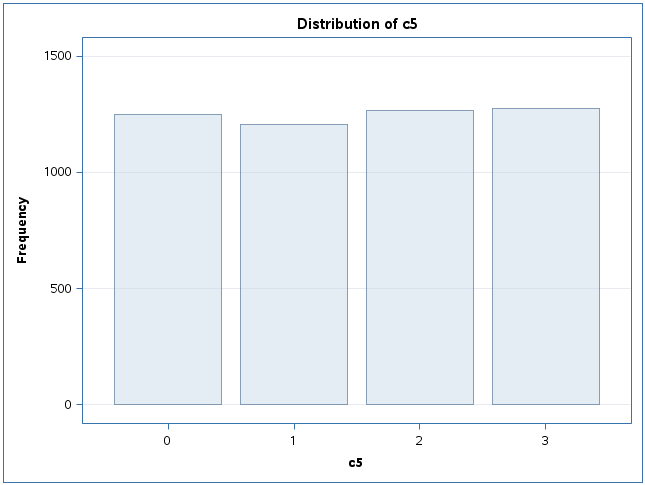
The histogram for the yTweedie dependent variable:



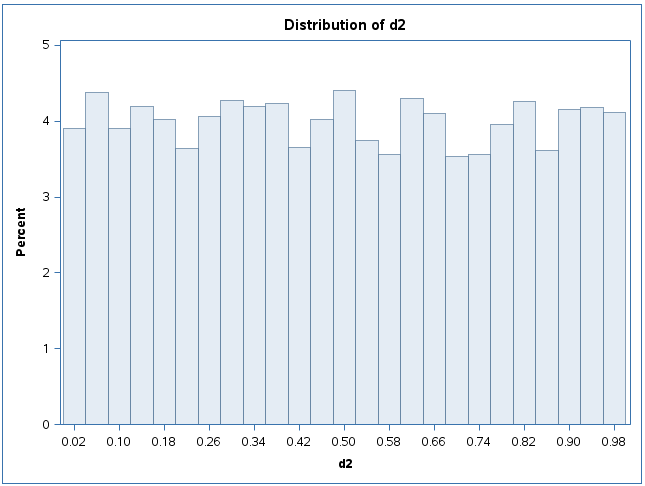
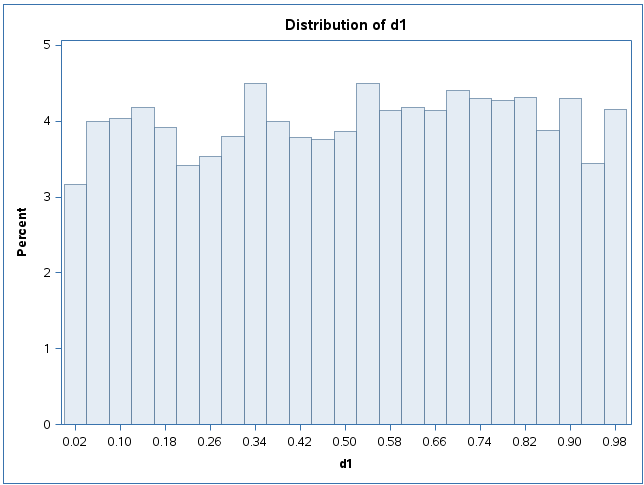
The independent character variable c1-c5:







The histogram for the two independent continuous variables d1 and d2:



The next lines contain the two SAS macros for the backwards elimination selection process using a Tweedie error function.

The first macro %MdStmt is a stand-alone macro. The main macro, %MdSelect, consists of multiple calls to the macro %MdStmt.

/\* Variable Selection Macro: Backwards elimination \*/

%let p=1.5;

options mlogic;

**%macro** MdStmt(

resvar = /\*response variable \*/

,expvar = /\*list of explanatory variables, separated by ' ' \*/

,clsvar = /\*classification variables in the CLASS statement separated by ' ' \*/

,p =

);

ods output Type3=pval(rename=source=parm);

proc genmod data=tmp1 NAMELEN=**50**;

if \_resp\_ > **0** then

d = **2**\*(\_resp\_\*(\_resp\_\*\*(**1**-&p)-\_mean\_\*\*(**1**-&p))/

(**1**-&p)-(\_resp\_\*\*(**2**-&p)-\_mean\_\*\*(**2**-&p))/(**2**-&p));

else d = **2**\* \_mean\_\*\*(**2**-&p)/(**2**-&p);

variance var = \_mean\_\*\*&p;

deviance dev = d;

class &clsvar;

model &resvar = &expvar /link=log type3 scale=pearson;

\*scwgt expos;

title "&resvar = &expvar";

run;

ods output close;

**%mend** MdStmt;

There are five macro parameters in the macro %MdSelect: &VAR, &INTVAR, &CATVAR, &SLSTAY and &POWER:

* &VAR is the response variable which will be passed into &RESVAR when calling the macro %MdStmt;
* &INTVAR includes all the potential explanatory variables which will be passed into &EXPVAR in %MdStmt only forthe first call;
* &CATVAR contains all the categorical explanatory variables which will be passed into %CLSVAR in %MdStmt;
* &SLSTAY is the criteria for removing variable;
* and &POWER is the power parameter of the Tweedie distribution

**%macro** MdSelect(

var= /\*response variable \*/

,intvar= /\*initial explanatory variables for full model \*/

,catvar= /\*categorical explanatory variables \*/

,slstay= /\*criterion for removing variable \*/

,power=

);

%let var=%upcase(&var);

%let intvar=%upcase(&intvar);

%let catvar=%upcase(&catvar);

%let power =&power;

%\*-------------------------------------------------------------------------\*;

%\* Create empty dataset "step" with only one column "parm". It will be \*;

%\* merged with "pval" from PROC GENMOD by "parm" \*;

%\*-------------------------------------------------------------------------\*;

proc sql;

create table step\_&var (parm char(**9**));

quit;

%\*------------------------------------------------------------------------------\*;

%\* %do %until performs multivariate backward model selection: \*;

%\* In each iteration: \*;

%\* 1. Run the logistic regression model \*;

%\* 2. Update the dataset "step\_&var" \*;

%\* 3. Create &pmax as the maximum p-value, and &varlist as the list of \*;

%\* variables without the one with the max p-value \*;

%\* 4. Check whether the max p-value <= &SLSTAY \*;

%\* 5. If NO, then eliminate the variable with max p-value, repeat step 1 to 4.\*;

%\* If YES, the loop stops \*;

%\*------------------------------------------------------------------------------\*;

%let i=1;

%do %until (&pmax<=&slstay);

%if &i = **1** %then

%***MdStmt***(resvar=&var ,expvar=&intvar, clsvar=&catvar, p=&power); %\*initial model;

%else %do;

%***MdStmt***(resvar=&var ,expvar=&varlist, clsvar=&catvar, p=&power); %\*reduced model;

%end;

proc sort data=step\_&var; by parm;

proc sort data=pval; by parm;

data step\_&var;

merge step\_&var pval;

by parm;

p&i=put(ProbChiSq, pvalue6.3);

drop ProbChiSq ChiSq DF;

run;

proc sql noprint;

select max(ProbChiSq) into :pmax

from pval;

select distinct parm into :varlist separated by ' '

from pval

having ProbChiSq^=max(ProbChiSq);

quit;

%let i=%eval(&i+1);

%end;

proc print data=step\_&var;

title "&var: model selection process";

run;

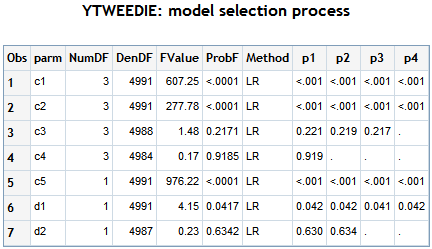
**%mend** MdSelect;

%***MdSelect***(var=yTweedie, intvar=c1 c2 c3 c4 c5 d1 d2, catvar=c1 c2 c3 c4 c5, slstay=**0.05**, power=**1.5**);

The execution of the above two macros create two outputs:

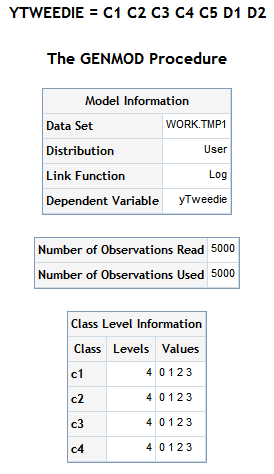
* A summary table of the model selection process
* The whole model selection process step by step

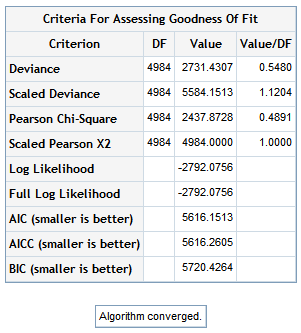
The summary table of the model selection process:

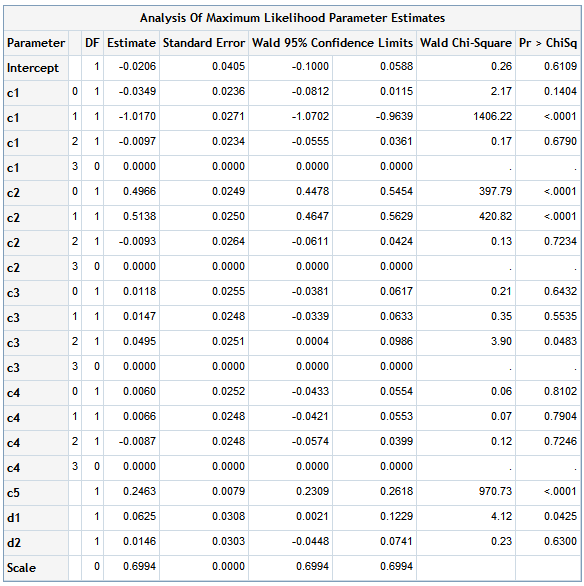


The table shows that the variable C4 is eliminated in the second step of the process. The variable D2 is eliminated in the third step. And the variable C3 is eliminated in the fourth step. After the fourth step the algorithm arrive at final main effects model.

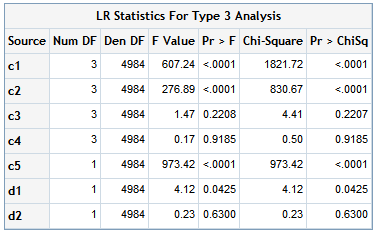
The whole variable selection process step by step:

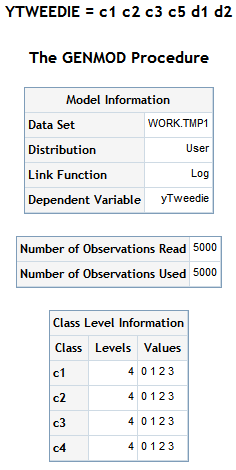


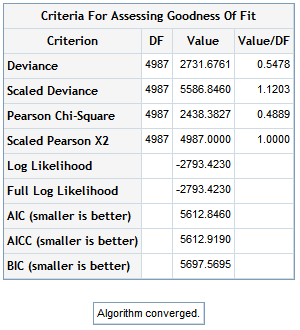


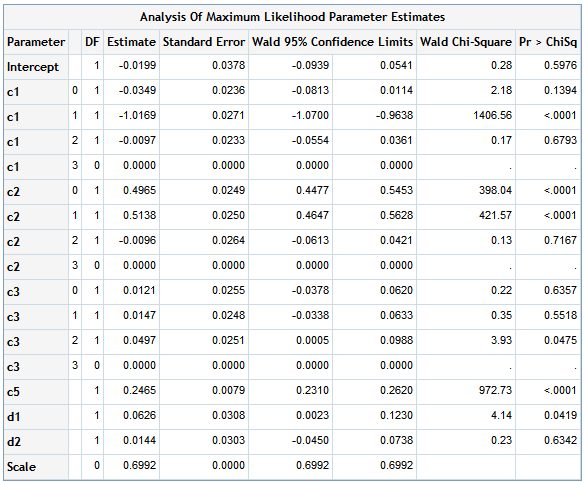




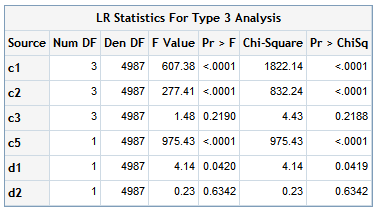


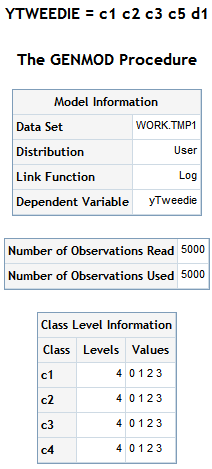


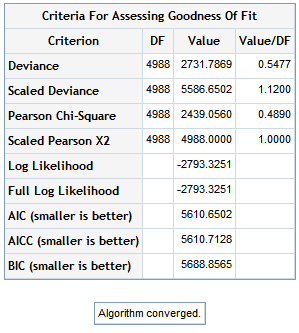


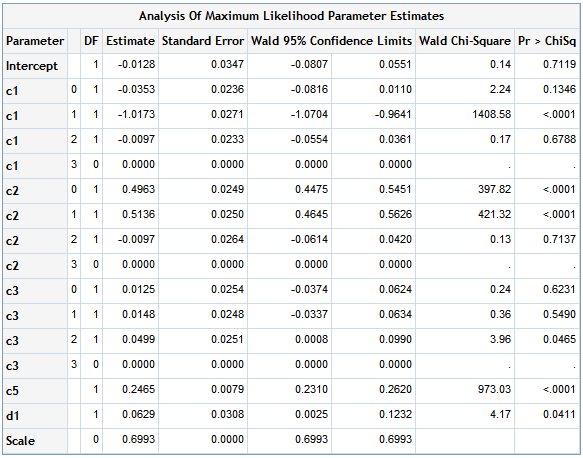




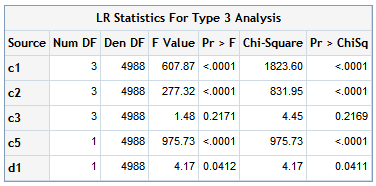


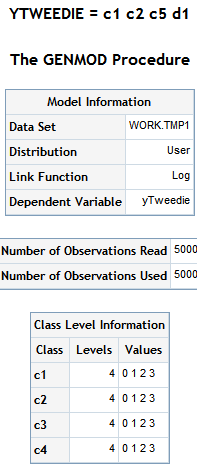


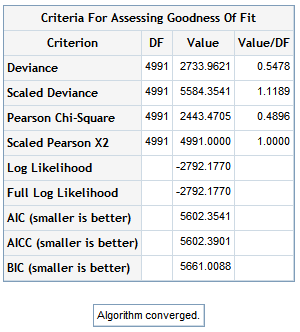


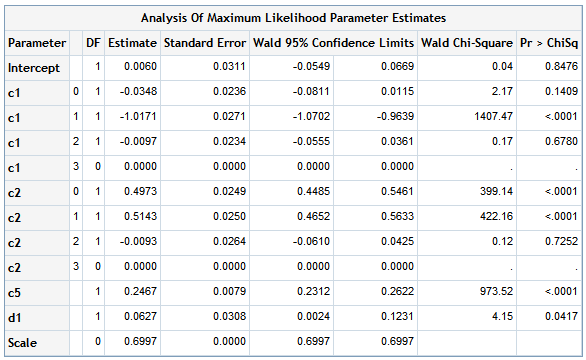




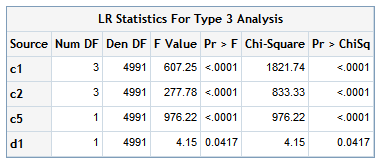












**Conclusion:**

The above lines shows how the variable selection algorithm eliminates those variables (C3, C4 and D2) no associated with the dependent variable yTweedie - remember that the illustrative dataset was arterially created with this aim. Therefore, the macro works accurately.

The SAS macros %MdStmt and %MdSelect:

* Performs a backwards elimination variable selection process
* The last step in the elimination process shows the selected model and a summary table of the elimination process
* The macro needs around 15 minutes to get results with a dataset of one million observations and around 13 variables
* The elimination criteria is based on the p-values of the type 3 analysis
* With small changes the macro is useful in a context with a GENMOD procedure under Gamma, Inverse Gaussian, Log-Normal, Binomial, Gaussian, Poisson, Negative Binomial, Zero Inflated Poisson and Zero inflated Negative Binomial error functions
* This macro could be useful as a template to create Forward and Stepwise variable selection processes
* One drawback of the backwards elimination process is that if the full model with all potential main factors does not converge the macro does not work. That is one of the reasons because a forward option is interesting
* The specification of the model is the same that the Tweedie macro used in the NAR project
* This macro only admits main factors. So, it is not possible to include interactions in the model statement of the GENMODE procedure. To include interactions it is needed create a new variable with the interaction

**References:**

A detailed explanation of the algorithm and the code appears here:

[Using Macro and ODS to Overcome Limitations of SAS® Procedures Jing Su and Wei (Lisa) Lin, Merck & Co, Inc., North Wales, PA](https://www.evernote.com/shard/s329/res/dc512c77-06e7-43ce-b66d-919b642a351a/cc26.pdf)

The dataset for the example comes from here:

<http://support.sas.com/documentation/cdl/en/statug/68162/HTML/default/viewer.htm#statug_genmod_examples12.htm>

I made some changes in order to get coherence results.