Workshop on Statistical modelling with SAS – part II

Building a model (three more aspects)

Dragos Calitoiu, Ph.D.

Hasan Mytkolli, Ph.D.

June 22, 2009

Hasan Mytkolli and Dragos Calitoiu are modellers for Bank of America – Canada Card Services. Opinions, ideas, and everything said in this presentation are of the authors and should not be confounded with their Position in the Bank;

Hasan Mytkolli and Dragos Calitoiu are also members of CORS (Canadian Operational Research Society) and OPTIMOD Research Institute.

Contact: Hasan.Mytkolli@mbna.com and Dragos.Calitoiu@mbna.com

The content of this presentation

- Finding the best subset of variables for a model;
- Checking for normality;
- The rationale of variable transformations.

Finding the best subset of variables

- Statistical model:

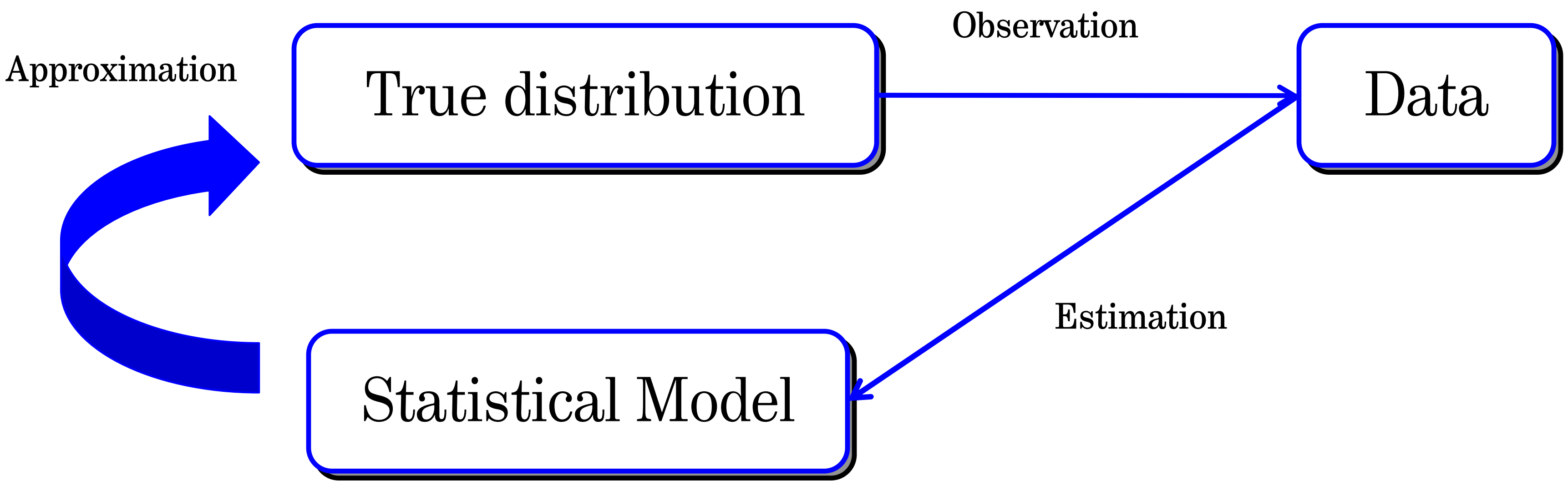
  - a set of mathematical equations which describe the behavior of an object of study in terms of random variables and of their associated probability distributions.

  - a probability distribution that uses observed data to approximate the true distribution of probabilistic events

- The mystery in building a statistical model is that the **true** subset of variables defining the **perfect** model is not known. The goal: to fit a **parsimonious** model that explains variation in the dependent variable with a small set of predictors, **as accurately as possible**.

- **True** vs. **parsimonious** model ;
- **Perfect** vs. **as accurately as possible**.

Finding the best subset of variables



For example:

In fitting a regression model, we want to detect the "true set of explanatory variables". However, due to computational constraints or business constraints, we want to use only 10-15 variables as the "true set". Which ones?

Finding the best subset of variables

- When the variables are **not** correlated with each other (rare):
  we can assess a variable's unique contribution and rank the variables.

- When the variables are correlated with each other (often):
  - we can assess a variable's relative importance with respect to the presence of other variables.
  - the variables interact with each other such that their total effect on the model's prediction is greater than the sum of their individual effects.
  - the Wald chi-square (in the case of logistic regression) is an indicator of a variable's relative importance and of selecting the best subset.

Finding the best subset of variables (the case of logistic regression)

Selecting the best subset process:

1. If you have the dataset, use your experience as a guide to identify all of the useful variables. Occasionally, you can build new variables (ratio, min, max, aggregate).
2. If there are too many variables, run a “survey” and produce a starter set.
3. Perform logistic regression analysis on the starter set; delete one or two variables with minimum Wald chi-square. After deleting, the order of the remaining variables can change. The greater the correlation between the remaining variables, the greater the uncertainty of declaring important variables.
4. Eliminate correlations.

Finding the best subset of variables (the case of logistic regression)

Selecting the best subset:

1. If you have the dataset, use your experience as a guide to identify all of useful variables.
2. If there are too many variables, run a "survey" and produce a starter set.
3. Perform logistic regression (automated procedure):
   - Backward Elimination (Top down approach);
   - Forward Selection (Bottom up approach);
   - Stepwise Regression (Combines Forward/Backward).
4. Eliminate correlations:
   - correlation;
   - variance inflation factor (VIF) ;
   - multicollinearity.

Finding the best subset of variables: **“the survey”**

```
%let varlist= var1 var2 var3 var4 var5;
%macro var_filt(inputfile, depend_var, nboots, bootsize, slent, slst, outfile);
   %do i=1 %to &nboots;
      proc surveyselect method=srs data=&inputfile out=boot&i    seed=%sysevalf(1000+&i*10) n=&bootsize;
      run;
      proc logistic data=boot&i desc noprint outest=log_var_filt_&i ;
                 model &depend_var=&varlist
                 /
                 selection=stepwise
                 slentry=&slent
                 slstay=&slst;
      run;
      proc datasets nolist; append data=log_var_filt_&i base= &outfile force; run;


   %end;
%mend var_filt;


options mprint mlogic spool;
%var_filt(file_name, dep_var, 20, 30000, 0.2, 0.1, file_output);
```

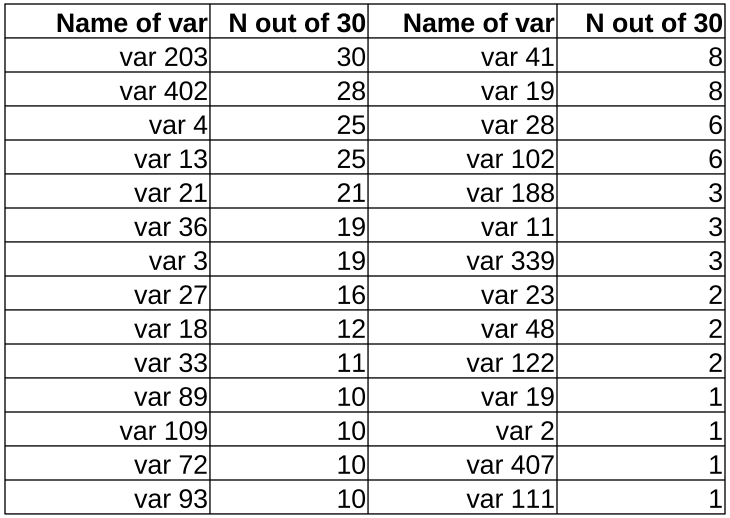
Finding the best subset of variables: **“the survey”**

```
%let varlist= var1 var2 var3 var4 var5; /* enter all the variables here */
%macro var_filt(inputfile, depend_var, nboots, bootsize, slent, slst, outfile);
   %do i=1 %to &nboots; /* run the stepwise logistic regression nboots time */
      proc surveyselect method=srs data=&inputfile out=boot&i
      seed=%sysevalf(1000+&i*10)  /* generate randomly a small data set for each run */
      n=&bootsize; run;
            proc logistic data=boot&i desc noprint outest=log_var_filt_&i ;
                 model &depend_var=&varlist
                 /
                 selection=stepwise
                 slentry=&slent /* the threshold of entering a variable into the model */
                 slstay=&slst;  /* the threshold of leaving the model */
      run;
      proc datasets nolist; append data=log_var_filt_&i base= &outfile force; /* append all the output files */
     run;

   %end;
%mend var_filt;

options mprint mlogic spool;
%var_filt(file_name, dev_var, 20, 30000, 0.2, 0.1, file_output);
```

```
 ods html file='var_selection.html'; title 'Variable Selection Logistic';

 proc means data=file_output n; run; title;

ods html close;
```

Finding the best subset of variables: **“the survey”**

Example - the output of the survey :



Name of var	N out of 30	Name of var	N out of 30
var 203	30	var 41	8
var 402	28	var 19	8
var 4	25	var 28	6
var 13	25	var 102	6
var 21	21	var 188	3
var 36	19	var 11	3
var 3	19	var 339	3
var 27	16	var 23	2
var 18	12	var 48	2
var 33	11	var 122	2
var 89	10	var 19	1
var 109	10	var 2	1
var 72	10	var 407	1
var 93	10	var 111	1

From a set of 460 variables, only 28 entered into the survey (namely they are independent variables in stepwise regressions). We decided to use only 14 of them (with the frequency greater than 10 out of 30 runs).

Remark: this is ONLY an example. Usually, we can continue to work with circa 40 variables, after this survey.

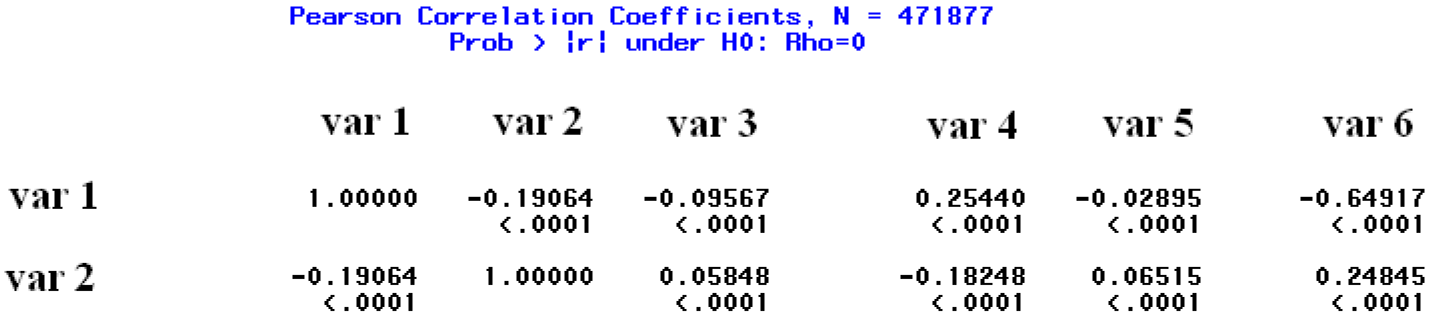
Finding the best subset of variables (the case of logistic regression)

Selecting the best subset:

1. If you have the dataset, use your experience as a guide to identify all of useful variables.
2. If there are too many variables, run a “survey” and produce a starter set.
3. Perform logistic regression (automated procedure):
    - Backward Elimination (Top down approach);
    - Forward Selection (Bottom up approach);
    - Stepwise Regression (Combines Forward/Backward).
4. Eliminate correlations:
    - correlation;
    - variance inflation factor (VIF) ;
    - multicollinearity.

Finding the best subset of variables: **Correlation**

proc corr data=filename; var var1 = var2 …varN; **run;**



```
                    Pearson Correlation Coefficients, N = 471877
                             Prob > |r| under H0: Rho=0

              var 1       var 2       var 3          var 4       var 5          var 6

var 1       1.00000    -0.19064    -0.09567        0.25440    -0.02895       -0.64917
                         <.0001      <.0001         <.0001      <.0001         <.0001

var 2      -0.19064     1.00000     0.05848       -0.18248     0.06515        0.24845
             <.0001                  <.0001         <.0001      <.0001         <.0001
```

- This procedure computes the Pearson correlation coefficient and produces simple descriptive statistics (mean, standard deviation sum, minimum and maximum) for all pairs of variables listed in the VAR statement.

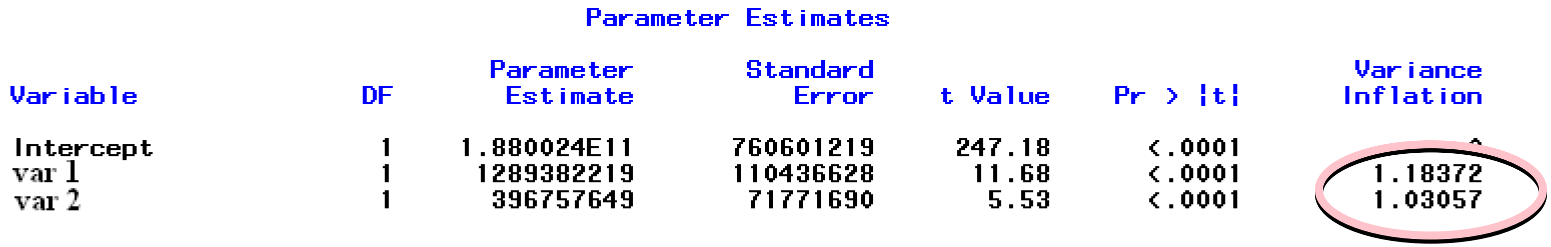
- It also computes a p-value for testing whether the true correlation is zero.

- The correlations are given in matrix form.

- A correlation of more than 0.7 between 2 variables encourages deleting one of the variables. Use experience and the Wald chi-square for identifying which one to delete.

Finding the best subset of variables: **VIF (variation inflation factor)**

proc reg data=filename; model var1 = var2 …varN/vif; **run;**



```
                                Parameter Estimates

                                 Parameter        Standard                               Variance
Variable                DF        Estimate           Error    t Value    Pr > |t|       Inflation

Intercept                1    1.880024E11       760601219     247.18      <.0001               0
var1                     1     1289382219       110436628      11.68      <.0001         1.18372
var2                     1      396757649        71771690       5.53      <.0001         1.03057
```

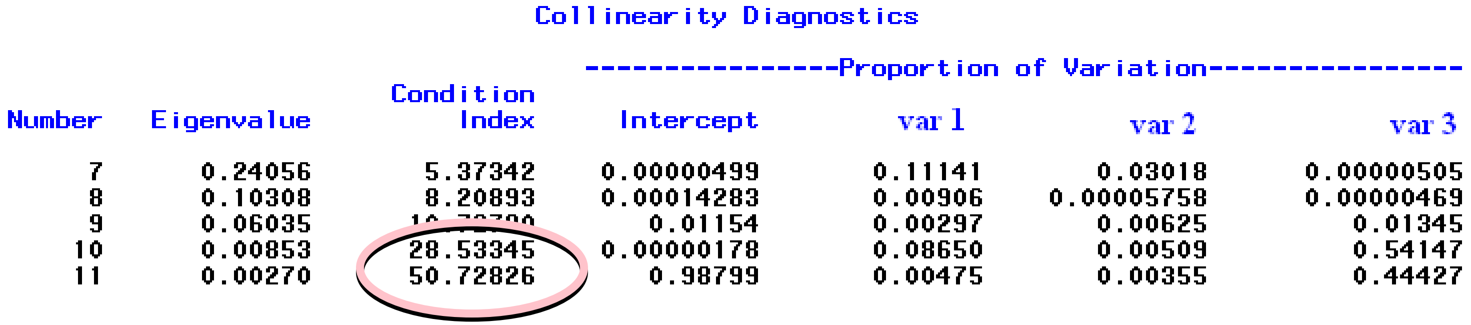
- Measure of how highly correlated each independent variable is with the other predictors in the model. Used to identify multicollinearity.

- Values larger than 10 for a predictor imply large inflation of standard errors for regression coefficients due to this variable being in the model.

- Inflated standard errors lead to small t-statistics for partial regression coefficients and wider confidence intervals.

Finding the best subset of variables: **Multicollinearity**

proc reg data=filename; model var1 = var2 …varN/collin; **run;**



```
                              Collinearity Diagnostics

                                        ----------------Proportion of Variation----------------
                         Condition
Number    Eigenvalue         Index    Intercept        var 1         var 2          var 3

     7       0.24056       5.37342   0.00000499      0.11141       0.03018     0.00000505
     8       0.10308       8.20893   0.00014283      0.00906    0.00005758     0.00000469
     9       0.06035    10.7[illegible]      0.01154      0.00297       0.00625        0.01345
    10       0.00853      28.53345   0.00000178      0.08650       0.00509        0.54147
    11       0.00270      50.72826      0.98799      0.00475       0.00355        0.44427
```

- For no collinearity between variables, the condition index must be less than 10.

- If this condition is not satisfied, you can find a pair where the proportion of variation is greater than 0.7. You have to eliminate one variable from that pair.

Remark: The intercept can be one member of the pair. You will delete the second member. You can conclude that a variable highly correlated with the intercept has a vary small variance, namely it is almost constant.

Finding the best subset of variables for the model

How many variables?

- A too small set of variables: a poor predicting response.
- Overfitted model (too many variables): a “too perfect” picture of the development data set (memorization) instead of capturing the desired pattern from the entire data set.

  Capitalizing on the idiosyncrasies of the training data.
- A well-fitted model is typically defined by a handful of variables because it does not include “idiosyncracy” variables.

Finding the best subset of variables for the **logistic** model

How good is a model?

Hirotugu **Akaike** (1971): AIC (Akaike's Information Criterion)

- Based on entropy, information and likelihood.

Informational statistical modelling:

- *Information criteria* are used to estimate expected entropy (analogous to expected uncertainty in the data) by taking simultaneously into account both the "goodness-of-fit" of the model and the complexity of the model required to achieve that fit.
- *Model complexity* involves both the number of parameters and the interaction (correlation) between them.

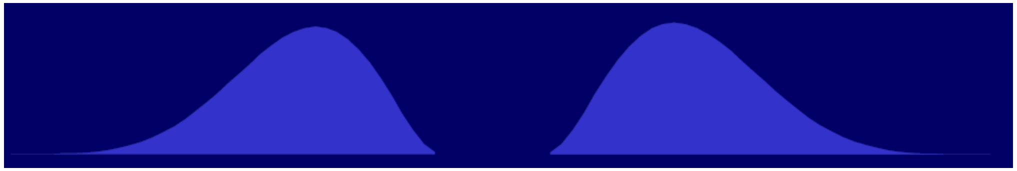
Testing for normality

- One of the critical steps in data analysis is checking for normality, in order to make assumptions about the distributions of the data.
- If the specified distributional assumption about the data is not valid, the analysis based on this assumption can lead to incorrect conclusions.

Testing for normality

- Skewness: the standardized 3rd central moment of a distribution;
- Positive skewness indicates a long right tail;
- Negative skewness indicates a long left tail;
- Zero skewness indicates symmetry around the mean;

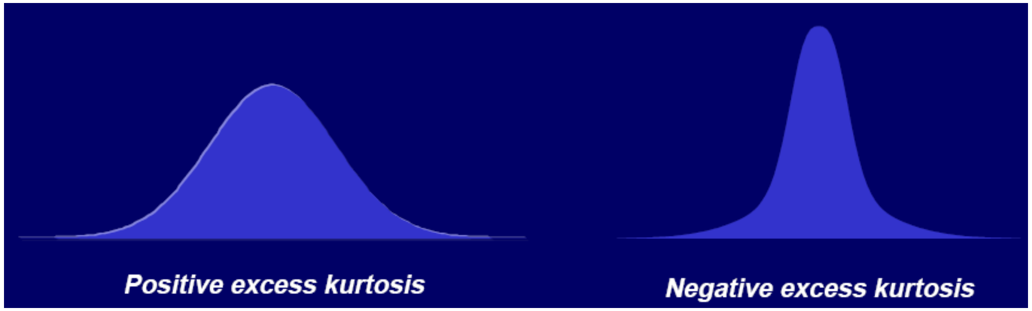
$$s = \frac{E(X - \mu)^3}{\sigma^3}$$



- Kurtosis: the standardized 4th central moment of a distribution;
- The kurtosis for the normal distribution is 3;
- Positive excess kurtosis indicates flatness (long, fat tails);
- Negative excess kurtosis indicates peakedness;

$$\kappa = \frac{E(X - \mu)^4}{\sigma^4}$$

$$excess\_\kappa = \kappa - 3$$



Testing for normality

- We used three well-known tests for rejecting the normality hypothesis, namely:
  1. Kolmogorov-Smirnov (K-S) test;
  2. Anderson-Darling test;
  3. Cramer-von Mises test.

G. Peng, *Testing Normality of Data using SAS* , PharmaSUG 2004, May 22-26, 2004, San Diego, California.

Testing for normality

1. **Kolmogorov-Smirnov test** :

- based on the empirical distribution function (EDF).;
- K-S D is the largest vertical distance between the distribution function F(x) and the EDF(F(x)) which is a step function that takes a step of height 1/n at each observation;
- to test normality, the K-S D is computed using the data set against a normal distribution with mean and variance equal to the sample mean and variance.

2. **Anderson-Darling test** :

- uses the quadratic class EDF, based on the squared difference (F(n)-F(x)^2 ;
- gives more weights to the tails than does the K-S test, the former being more sensitive to the center of the distribution than at the tails;
- The procedure of emphasizing the tails is done by setting a function weighting the square difference.

3. **The Cramer-von Mises test** :

- also based on the quadratic class of EDF;
- however, the weight function in this scenario is considered equal to unity.

The last two tests describe better the entire data set from the sum of the variances point of view.

The K-S test is much sensitive to the anomalies in the sample.

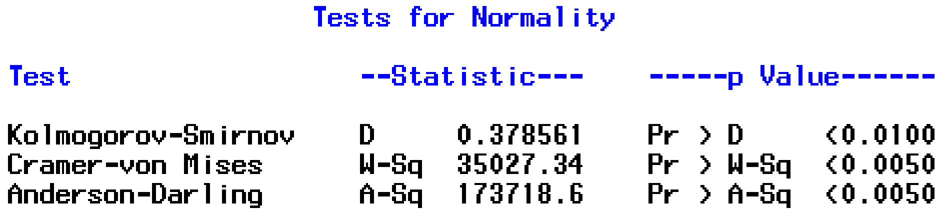
Testing for normality

- SAS has implemented the commonly used normality tests in PROC UNIVARIATE.

```
proc univariate  data=name  normal  plot  vardef=df;  var name;
```

- The **normal** option computes the three tests of the hypothesis that the data comes from a normal population. The p-value of the test is also printed.

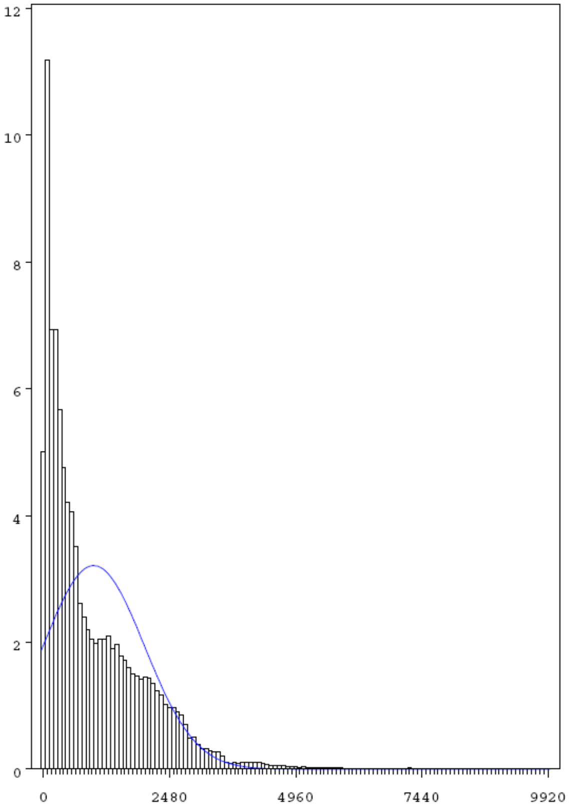
- The **vardef**= option specifies the divisor used in computing the sample variance.
  vardef=df, the default, uses the degrees of freedom n-1;
  vardef=n uses the sample size n.

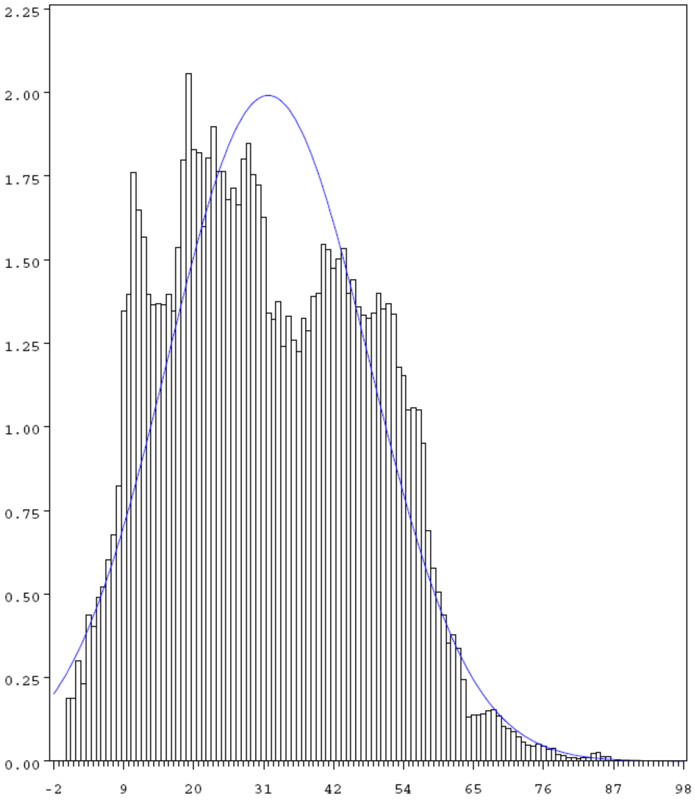


```
                         Tests for Normality

Test                  --Statistic---    -----p Value------

Kolmogorov-Smirnov    D     0.378561    Pr > D      <0.0100
Cramer-von Mises      W-Sq  35027.34    Pr > W-Sq   <0.0050
Anderson-Darling      A-Sq  173718.6    Pr > A-Sq   <0.0050
```

An alternative procedure is PROC CAPABILITY.

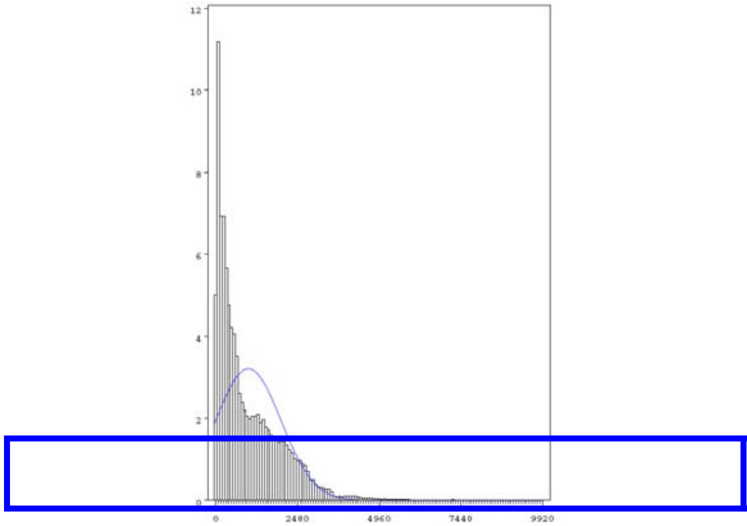
Testing for normality

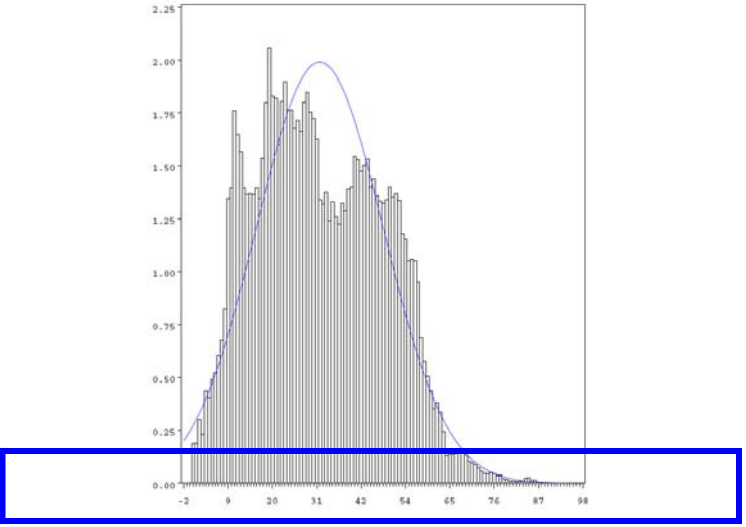




- For the left graph: mean 992.18, std 994.70.,skewness 1.53, kurtosis 3.29;
- For the right graph: mean 5.50, std 2.87, skewness -9.29E-7, kurtosis -1.22;
- For the normal distribution: skewness = 0, kurtosis = 3.

Testing for normality





Test	Statistic		p Value		Statistic		p Value	
Kolmogorov-Smirnov	D	0.16	Pr>D	<0.010	D	0.05	Pr>D	<0.010
Cramer-von Mises	W-Sq	6295.14	Pr>W-Sq	<0.005	W-Sq	845.09	Pr>W-Sq	<0.005
Anderson-Darling	A-Sq	36465.51	Pr>A-Sq	<0.005	A-Sq	5189.2	Pr>A-Sq	<0.005

Transforming data

- Data transformation is one of the remedial actions that may help to make data normal.

- The rationale of variable transformations.
  1. To reduce variance;
  2. To transform optimally the information content hidden in data.

Transforming data: Ladder of Powers and Bulging Rule

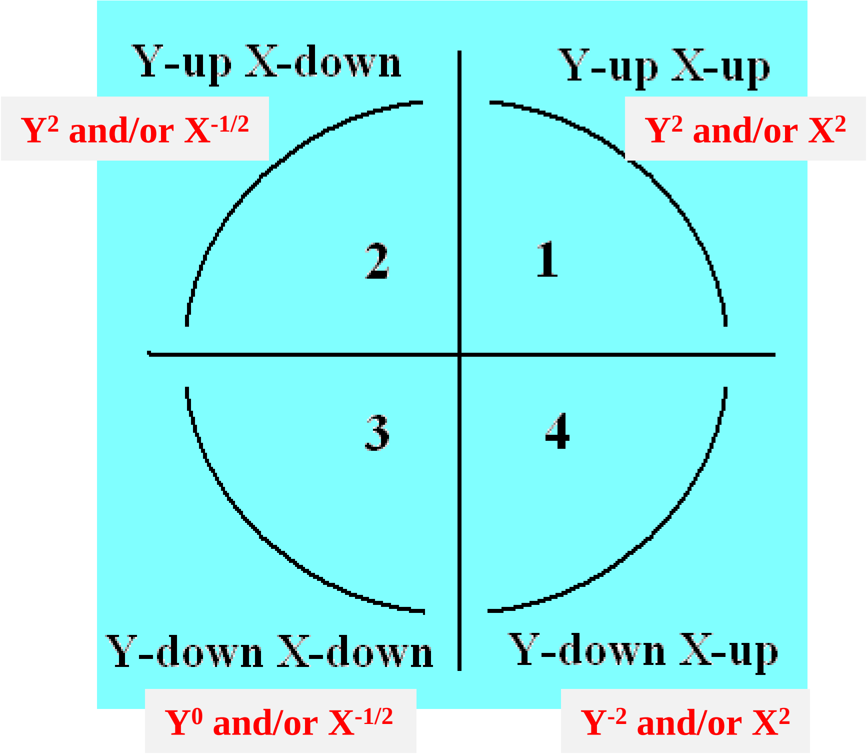
- A method of re-expressing variables to straighten a relationship between two continuous variables, say X and Y. Examples are:

  - $1/x^2$ reciprocal of square (power -2);
  - $1/x$ reciprocal (power -1);
  - $x^0=\ln(x)$ natural logarithm;
  - $x^{½}$ square root (power 0.5);
  - x^2 square (power 2).

- Going up-ladder of powers: raising a variable to a power greater than 1: X^2, X^3…or Y^2 ,Y^3…

- Going down-ladder of powers: raising a variable to a power less than 1: $X^{1/2}$,X^0, $X^{-1/2}$ or $Y^{1/2}$, Y^0,$Y^{-1/2}$

- Remarks: Not so much freedom with the dependent variable ; if you have $Y=f(X_1,X_2)$ you cannot do $Y^{1/2}=G(X_1)$ and $Y^0=H(X_2)$;



The Box-Cox Transformations

- George Box and David Cox developed a procedure to identify an appropriate exponent (λ) to transform data into a "normal shape". The exponent λ indicates the power to which all data should be raised.

- The aim of the Box-Cox transformations is to ensure that the usual assumptions for the Linear Model hold, namely **y** ~N(**X**β, $\sigma^2 I_n$).

- The original for

$$y(\lambda) = \begin{cases} \frac{y^\lambda - 1}{\lambda}, & \text{if } \lambda \neq 0; \\ \log y, & \text{if } \lambda = 0. \end{cases}$$

Practically, the algorithm searches from λ=-5 to λ=+5 with a step of 0.2 until the **best** value is found.

- Weakness: Cannot be applied to negative numbers.

The Box-Cox Transformations (extended)

- They proposed in the same paper an extension.
- Find λ_2 such that $y+\lambda_2>0$ for any y. With this condition, the transformation can accommodate negative y's:

$$y(\boldsymbol{\lambda}) = \begin{cases} \frac{(y+\lambda_2)^{\lambda_1}-1}{\lambda_1}, & \text{if } \lambda_1 \neq 0; \\ \log(y+\lambda_2), & \text{if } \lambda_1 = 0. \end{cases}$$

Yeo and Johnson Transformations

- Yeo and Johnson (2000) proposed a new set of transformations for real numbers:

$$y(\lambda) = \begin{cases} \frac{(y+1)^{\lambda}-1}{\lambda}, & \text{if } \lambda \neq 0,\ y \geq 0; \\ \log(y+1), & \text{if } \lambda = 0,\ y \geq 0; \\ \frac{(1-y)^{2-\lambda}-1}{\lambda-2}, & \text{if } \lambda \neq 2,\ y < 0; \\ -\log(1-y), & \text{if } \lambda = 2,\ y < 0. \end{cases}$$

The code

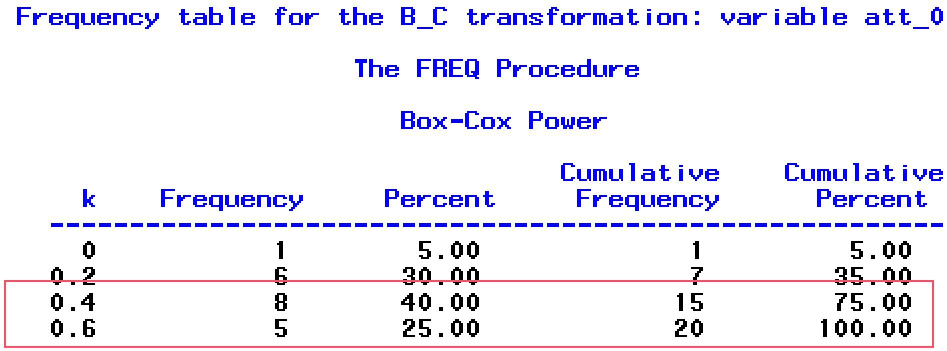
```
options mlogic mprint ;
     /** This is the main macro code **/
     /** Fileinput - is your data file **/
     /** ul - is the upper limit for the power transformation **/
     /** ll - is the absolute value of the lower limit for the power transformation          **/
     /** base - is the base choice or alternative to which other alternatives will compare to**/
     /** dependent - is the dependent variable  **/
     /** independent - is the independent variable you are looking for transformation        **/

%macro boot_box_cox(fileinput, bootvol, nrboots,dependent, independent, lowpower, highpower, step);

       data temp (keep=&dependent &independent); set &fileinput; run;
       proc datasets nolist; delete histo_data; run;
       %do i=1 %to &nrboots;
proc surveyselect data=temp method=srs out=boot&i seed=%sysevalf(10000+&i*10)    n=&bootvol;
               run;
          data tempi (rename=(&independent=x)); set boot&i; run;
          data tempi; set tempi;
           if (x<0) then do;
                    %do k10=%sysevalf(0) %to %sysevalf(-&lowpower.*10) %by %sysevalf(&step.*10);
                    k=%sysevalf(-&k10/10); y=(1-x); z=2-k;
                    xk=((1-x)**(2-k)-1)/(k-2);
                    output;
                    %end;
              %do k10=%sysevalf(&step.*10) %to %sysevalf(&highpower.*10) %by %sysevalf(&step.*10);
                    k=%sysevalf(&k10/10);
                    %if %sysevalf(&k10.) = 20 %then %do;
                        xk=-log(1-x);
                        output;
                        %end;
                   %else %do;
                        xk=((1-x)**(2-k)-1)/(k-2);
                        output;
                    %end;
               %end;
                   end;
              else do;
```

The search for finding the optimum exponent is done in the range of (-lowerpower, highpower) with a fixed step.

The code…

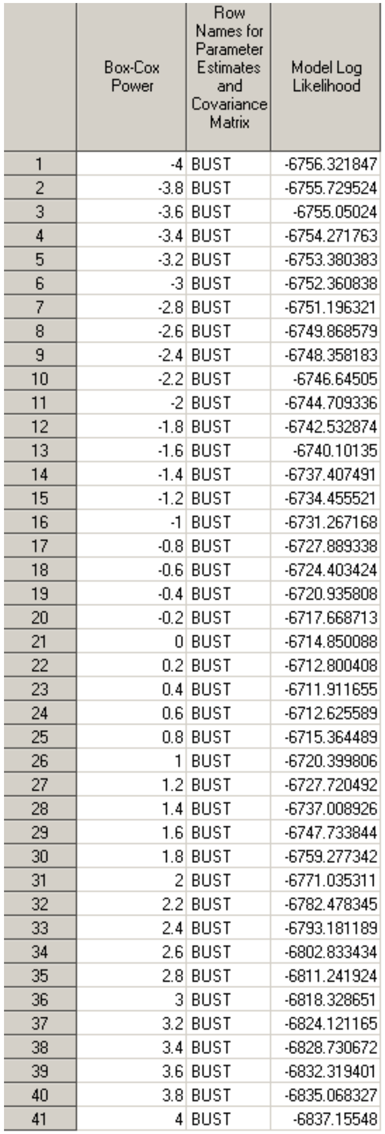
```
%do k10=%sysevalf(&step.*10) %to %sysevalf(-&lowpower.*10) %by %sysevalf(&step.*10);
                  k=%sysevalf(-&k10/10);
                  xk=((x+1)**k-1)/k;
                  output;
                  %end;
                  k = 0;
                  xk=log(x+1);
                  output;

           %do k10=%sysevalf(&step.*10) %to %sysevalf(&highpower.*10) %by %sysevalf(&step.*10);
                  k=%sysevalf(&k10/10);
                  xk=((x+1)**k-1)/k;
                  output;
              %end;
                 end;
       label k='Box-Cox Power' ;
       run;
       proc sort data=tempi; by k; run;
    /** You have created all the transformed data ready for one-variable estimation      **/

   proc logistic data=tempi descending noprint outest=boot_k&i (keep=_name_ k _lnlike_);
         model &dependent= xk; by k;
          run;
      title;
      proc gplot data=boot_k&i;
                  symbol1 v=none i=join;
                  plot _lnlike_*k;
      title "BC Utilization likelihood - &independent i=&i "; run; title;
proc sort data= boot_k&i ; by descending _lnlike_; run;
       data tempout;  set boot_k&i ( obs=1); run;
       proc append base=histo_data data=tempout force;run;
       proc datasets nolist; delete tempi tempout boot&i    boot_k&i     ; run;
    %end;
       title4 "Frequency table for the B_C transformation: variable &independent";
    proc freq data=histo_data ;
      tables k; run; title4 ;
   %mend boot_box_cox;


    %boot_box_cox(file_name, 50000, 20, var_dep, var_indep, -4, 4, 0.2);

```

The output: the histogram and one file as example.



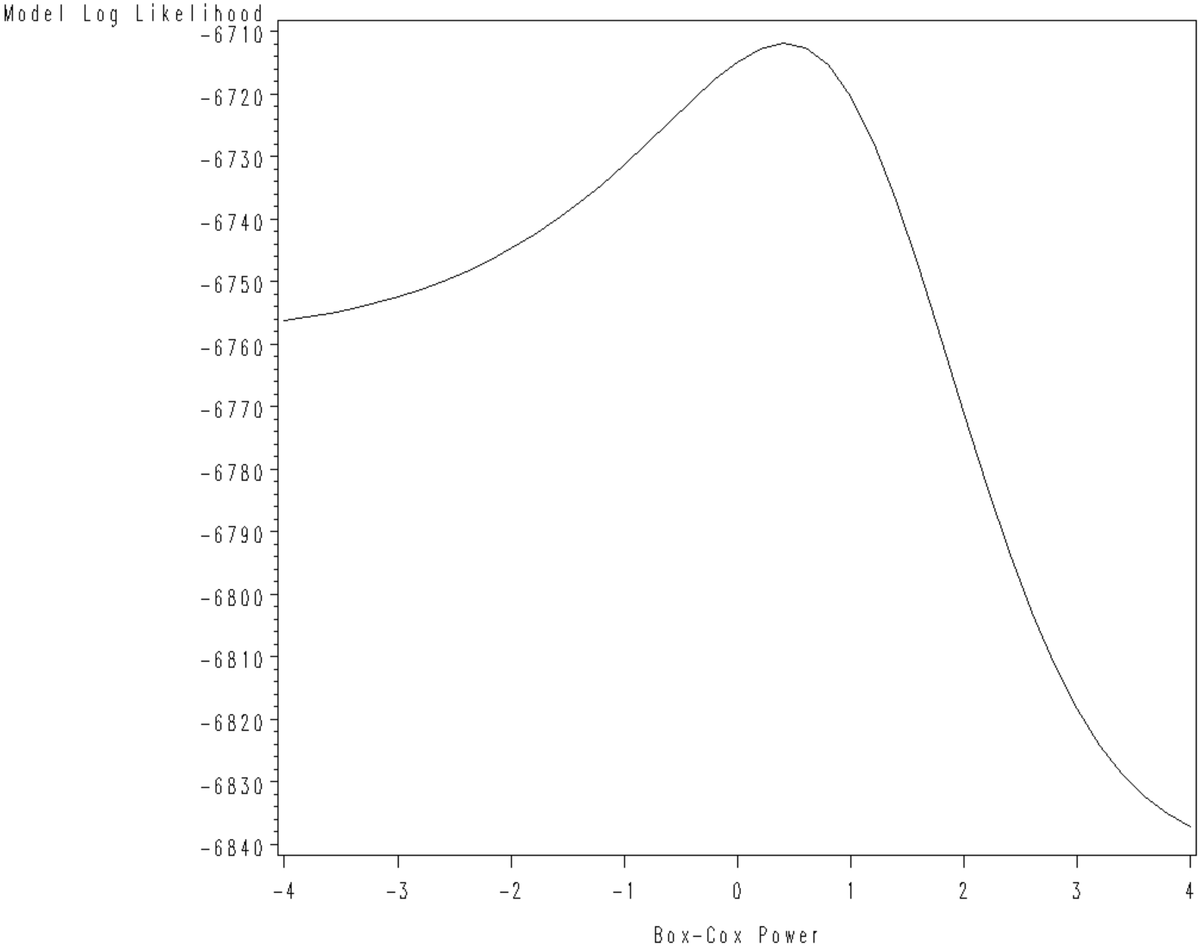
```
Frequency table for the B_C transformation: variable att_0

                 The FREQ Procedure

                   Box-Cox Power

                                            Cumulative    Cumulative
  k      Frequency      Percent              Frequency      Percent
--------------------------------------------------------------------
  0              1         5.00                      1         5.00
0.2              6        30.00                      7        35.00
0.4              8        40.00                     15        75.00
0.6              5        25.00                     20       100.00
```

```
rsubmit;
proc gplot data=boot_k&i;
                symbol1 v=none i=join;
                plot _lnlike_*k;
     title "BC Utilization likelihood - &independent
i=&i ";
     run;


endrsubmit;
```

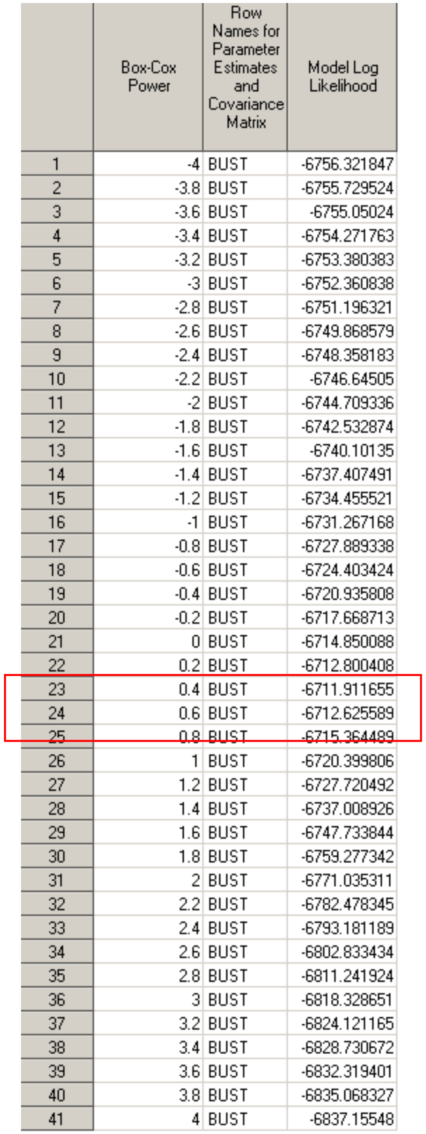


	Box-Cox Power	Row Names for Parameter Estimates and Covariance Matrix	Model Log Likelihood
1	-4	BUST	-6756.321847
2	-3.8	BUST	-6755.729524
3	-3.6	BUST	-6755.05024
4	-3.4	BUST	-6754.271763
5	-3.2	BUST	-6753.380383
6	-3	BUST	-6752.360838
7	-2.8	BUST	-6751.196321
8	-2.6	BUST	-6749.868579
9	-2.4	BUST	-6748.358183
10	-2.2	BUST	-6746.64505
11	-2	BUST	-6744.709336
12	-1.8	BUST	-6742.532874
13	-1.6	BUST	-6740.10135
14	-1.4	BUST	-6737.407491
15	-1.2	BUST	-6734.455521
16	-1	BUST	-6731.267168
17	-0.8	BUST	-6727.889338
18	-0.6	BUST	-6724.403424
19	-0.4	BUST	-6720.935808
20	-0.2	BUST	-6717.668713
21	0	BUST	-6714.850088
22	0.2	BUST	-6712.800408
23	0.4	BUST	-6711.911655
24	0.6	BUST	-6712.625589
25	0.8	BUST	-6715.364489
26	1	BUST	-6720.399806
27	1.2	BUST	-6727.720492
28	1.4	BUST	-6737.008926
29	1.6	BUST	-6747.733844
30	1.8	BUST	-6759.277342
31	2	BUST	-6771.035311
32	2.2	BUST	-6782.478345
33	2.4	BUST	-6793.181189
34	2.6	BUST	-6802.833434
35	2.8	BUST	-6811.241924
36	3	BUST	-6818.328651
37	3.2	BUST	-6824.121165
38	3.4	BUST	-6828.730672
39	3.6	BUST	-6832.319401
40	3.8	BUST	-6835.068327
41	4	BUST	-6837.15548

The output: the likelihood from one file (out of 20)



The Box_Cox index is 0.4 in this example.



	Box-Cox Power	Row Names for Parameter Estimates and Covariance Matrix	Model Log Likelihood
1	-4	BUST	-6756.321847
2	-3.8	BUST	-6755.729524
3	-3.6	BUST	-6755.05024
4	-3.4	BUST	-6754.271763
5	-3.2	BUST	-6753.380383
6	-3	BUST	-6752.360838
7	-2.8	BUST	-6751.196321
8	-2.6	BUST	-6749.868579
9	-2.4	BUST	-6748.358183
10	-2.2	BUST	-6746.64505
11	-2	BUST	-6744.709336
12	-1.8	BUST	-6742.532874
13	-1.6	BUST	-6740.10135
14	-1.4	BUST	-6737.407491
15	-1.2	BUST	-6734.455521
16	-1	BUST	-6731.267168
17	-0.8	BUST	-6727.889338
18	-0.6	BUST	-6724.403424
19	-0.4	BUST	-6720.935808
20	-0.2	BUST	-6717.668713
21	0	BUST	-6714.850088
22	0.2	BUST	-6712.800408
23	0.4	BUST	-6711.911655
24	0.6	BUST	-6712.625589
25	0.8	BUST	-6715.364489
26	1	BUST	-6720.399806
27	1.2	BUST	-6727.720492
28	1.4	BUST	-6737.008926
29	1.6	BUST	-6747.733844
30	1.8	BUST	-6759.277342
31	2	BUST	-6771.035311
32	2.2	BUST	-6782.478345
33	2.4	BUST	-6793.181189
34	2.6	BUST	-6802.833434
35	2.8	BUST	-6811.241924
36	3	BUST	-6818.328651
37	3.2	BUST	-6824.121165
38	3.4	BUST	-6828.730672
39	3.6	BUST	-6832.319401
40	3.8	BUST	-6835.068327
41	4	BUST	-6837.15548

Summary

- Finding the best subset of variables for a model;
- Checking for normality;
- The rationale of variable transformations.
- **QUESTIONS?**
- Contact: Hasan.Mytkolli@mbna.com and Dragos.Calitoiu@mbna.com