Modeleasy+ Sample Session

To use Modeleasy+ to analyze a model, there are a few simple steps to follow. You must get the model and data into the Modeleasy+ workspace, estimate the coefficients of the model, simulate the model and finally, examine the results of the analysis.

The following links will describe and demonstrate these steps in a sample Modeleasy+ session:

- Analyze a model
  - -MDL specification of the Klein1 Model
  - -MDL keywords
- Getting the Model into the Modeleasy+ workspace
- Getting the Data into the Modeleasy+ workspace
  - -The GETLIST command
  - -The GETFAME command
  - -Fortran I/O
- Estimate the Coefficients
  - -Output interpretation
  - -Output options
- Modeleasy+ Simulation
  - -The BUILDMODEL command
  - -The SIMULATE command
- Examining Results

Analyze a Model

To use Modeleasy+ to analyze a model, there are a few simple steps to follow and these sections will guide you through the process.

It should be noted that all of the analysis presented are based on the **`Klein1`** model, which is shown below written in the **`Model Description Language (MDL)`** of Modeleasy+.

MDL Specification of the Klein1 Model

```
BEHAVIORAL> C TSRANGE 1921 1 1941 1
EQ> C = A1 + A2*P + A3*LAG(P,1) + A4*(W1+W2)
COEFF> A1 A2 A3 A4
DEFINECOEFF> COE(1)

BEHAVIORAL> I TSRANGE 1921 1 1941 1
EQ> I = B1 + B2*P + B3*LAG(P,1) + B4*LAG(K,1)
COEFF> B1 B2 B3 B4
DEFINECOEFF> COE(5)

BEHAVIORAL> W1 TSRANGE 1921 1 1941 1
EQ> W1 = C1 +C2*(Y+T-W2) + C3*LAG(Y+T-W2,1) + C4*TIME
COEFF> C1 C2 C3 C4
DEFINECOEFF> COE(9)

IDENTITY> Y
EQ> Y = C + I + G - T

IDENTITY> P
EQ> P = Y - (W1+W2)

IDENTITY> K
EQ> K = LAG(K,1) + I
```

MDL Keywords

The model as written above, demonstrates several features of the **`MDL`** which should be mentioned. As discussed in the description of the **`MDL`**, the model's equations and characteristics are identified by keywords of the form, **`KEY_WORD>`**. The type of equation is identified by the keywords **`BEHAVIORAL>`** and **`IDENTITY>`**, while the actual form of the equation follows the **`EQ>`** keyword. In the behavioral equations, the coefficients in the equation are identified by the **`COEFF`**> keyword, all other names being treated as variables. Finally, the workspace location where the values of the coefficients will be saved by estimation and retrieved by simulation is specified in the **`DEFINECOEFF`**> keyword.

Once the model has been formulated, it can be saved in a library for later use. In the samples that follow, it is assumed that this has been done.

Click on the links below to follow the steps to use Modeleasy+ for analyzing a model:

- Get the model into the Modeleasy+ workspace;
- Get the data into the Modeleasy+ workspace;
- Estimate the coefficients of the model;
- Simulate the model;
- Examine the results of the analysis.

Getting the model into the Modeleasy+ workspace

In order to analyze a model with Modeleasy+, the model and its associated data must be brought into the user's workspace. Some of the methods for doing this are discussed in the following sections.

Getting the Model

The model may be defined by using the editor in a Modeleasy+ session, an external editor or the Visual Model Editor utility program. The model is usually saved as an ASCII file in some directory of the user's choice. If this has been done, then the model may be placed in the user's Modeleasy+ workspace by the general command

```
:_getdeck model_name suffix suffix_name dir "dir_name"
```

where "**`model_name`**" is the name of the model, "**`suffix_name`**" is the suffix of the model's file name, and "**`dir_name`**" is the name of the directory where the user has saved the model. For example, if the **`Klein1`** model had been saved as **`c:\models\Klein1.dec`**, then the command would be

```
:_getdeck klein1 dir "c:\models"
```

and the model would be defined in the user's workspace. The suffix keyword has been omitted in the above command because the model has been saved with the default suffix name of "**dec**".

A library where the user saves models can be defined using the **`LIBRARY`** command so that the directory name need not be specified in subsequent uses. The commands

```
:_library models "c:\models"
:_getdeck klein1 library models
```

are equivalent to the previous command, the advantage being that the library definition can be saved from session to session.

Getting the data into the Modeleasy+ workspace

The data needed to estimate and simulate the model may be obtained by several methods. These include using the **`GET`** or **`GETLIST`** commands, a database command, or the **`Fortran I/O`** capabilities of Modeleasy+. When the data has been moved into the Modeleasy+ workspace by one of these methods, the **`GETLIST`** command is the most convenient for subsequent retrievals.

Getting the Data

For the Klein1 model, the data for the variables **`C, G, I, K, P, T, TIME, W1, W2`**, and **`Y`** must be in the user's workspace. When this has been done, then they may be saved as a group, called **`klein1v,`** using the commands below.

```
:_names
C, G, I, K, P, T, TIME, W1, W2, Y
:_klein1v = namelist(c,g,i,k,p,t,time,w1,w2,y)
:_library data "c:\data"
:_keeplist klein1v library data
```

where the **LIBRARY** command has been used to specify the directory where the data is to be saved.

The GETLIST command

To get the data into the user's workspace for subsequent uses, it is only necessary to enter the command

```
:_getlist klein1v library data
```

and the data grouped under the name **klein1v** will be retrieved.

The GETFAME command

To initially retrieve each of the individual pieces of data, the user may use a database command or the **Fortran I/O** capabilities of Modeleasy+. If the user has the **FAME** database installed and the time series are stored there, then the command

```
:_getfame(c,g,i,k,p,t,time,w1,w2,y : db_name)
```

could be used to retrieve the data for the **Klein1** model from the database named "**db_name**".

Fortran I/O

If the user has only ASCII files of the data, then these files can be read using the **Fortran I/O** capabilities of Modeleasy+. For example, suppose the file containing the data for C is named "**c:\temp.dat**" and contains data of the form

```
39.80 41.90 45.00 49.20 50.60
52.60 55.10 56.20 57.30 57.80
55.00 50.90 45.60 46.50 48.70
51.30 57.70 58.70 57.50 61.60
65.00 69.70
```

then this data may be read into the Modeleasy+ workspace using the commands below.

```
:_assign 10 "c:\temp\c.dat"
:_fmt = "(5(f5.2,1x))"
:_c = array(22:)
:_read 10 fmt : c
```

where it is known that there are 22 observations in the time series and so C is predefined to have 22 elements. Of course, this array of data must then be converted into a time series using the **TIMESERIES** command

```
:_c = timeseries(c:1920,1,1)
```

which defines C to be an annual time series starting in 1920.

Because the **Fortran I/O** method is rather cumbersome to repeat from session to session, it is recommended that the user save the model's group of data using the **KEEPLIST** command as discussed at the beginning of this section.

Once the model and data are in the user's workspace, then the analysis of the model can be undertaken.

Estimate the Coefficients

In order to estimate the coefficients in a model, both the model and its associated data must be present in the user's workspace. This has discussed previously and the commands below are used to accomplish this.

```
:_getdeck klein1 on models
:_getlist klein1v on data
```

Estimation output

With the model and data in the workspace, the only command needed to do an **OLS** estimation of the model is **ESTIMATE**. This use of this command is shown below along with a partial listing of the default output corresponding to the first equation in the **Klein1** model.

```
:_estimate klein1
*Linkule ESTIMATE. Version 5/1/98*
The number of behavioral equations to be estimated in block ONE is 3.
Behavioral Equation C in Block ONE
Ordinary Least Squares Estimation Technique

Behavioral Equation C
  C = 16.237
    ( 12.464 )
           +.19293 * P
            ( 2.1153 )
           +.089885 * LAG(P,1)
            ( .99158 )
           +.79622 * (W1+W2)
            ( 19.933 )

  R-Squared : .98101
  Adjusted R-Squared : .97766
  Durbin-Watson Statistic : 1.3675
  Sum of squares of residuals : 17.879
  Standard Error of Regression : 1.0255
  Log of the Likelihood Function :-28.109
  F-statistic ( 3 , 17 ) : 292.71
  F-probability : 5.9605E-8
  Mean of Dependent Variable : 53.995
  Number of Observations : 21
  Number of Degrees of Freedom : 17
  Current Sample : 1921 1 1941 1
```

Output interpretation

The output is displayed in such a way that the original equation is displayed with the coefficient names replaced by their estimated values i.e.,

```
C = A1 + A2 * P + A3 * LAG(P,1) + A4 * (W1+W2)
```

has become

```
C = 16.237 + .19293 * P + .089885 * LAG(P,1) + .79622 * (W1+W2)
```

for easier interpretation. Under each estimated coefficient is the **T-statistic** associated with that coefficient. Also displayed are the (default) statistical test results for the equation in order to assist the user in judging how well the equation has been estimated.

The results of the coefficient estimations have been saved in an array named **COE**, as defined with the **DEFINECOEFF>** keyword for each behavioral equation. The results of the estimation are defined in the workspace under this name as shown below

```
:_coe
COE (A 12 Component Array)
16.237 .19293 .089885 .79622 10.126 .47964 .33304 -.11179
1.497 .43948 .14609 .13025
```

where the first 4 values correspond to behavioral equation C, the next 4 to I, and the last 4 to W1. Since these results are defined just as any other object, they can be manipulated and saved in any way that the user wants. This is particularly useful when simulations are going to be done. The simulation will look for the estimated coefficients under this name, and so the object can simply be retrieved before the simulation without any need for re-estimating the model.

The output for each behavioral equation estimated follows this form. For large models, this can occupy quite a bit of space on the screen. There is a way to suppress the estimation output and that is controlled by the **OUTPUTOPTION** command. In the example below, the output is turned off and the **Klein1** model re-estimated.

```
:_outputoption noprint
*Linkule OUTPUTOP. Version 5/1/98*
:_estimate klein1
*Linkule ESTIMATE. Version 5/1/98*
 The solution for equation C has been derived.
 The solution for equation I has been derived.
 The solution for equation W1 has been derived.
```

Of course, if any errors had occurred during the estimation of a particular equation, messages describing the error would not have been turned off by this option.

Output options

An alternate syntax to the **ESTIMATE** command allows the user to estimate a single equation (or group of equations) in the model. For example, the command

```
:_estimate klein1 c
```

would estimate only equation C in the Klein1 model and the output would appear according to the **OUTPUTOPTION** setting.

Equation selection

Once the model has been estimated to the user's satisfaction, then the simulation of the model can be done.

Modeleasy+ Simulation

In order to simulate a model, the model, the data associated with the model, and the coefficients of the model must be in the user's workspace. The coefficients must be defined using the names found in the **DEFINECOEFF>** keywords of the model. If the coefficients have been obtained by using the Modeleasy+ **ESTIMATE** command, then this has been done automatically.

The Modeleasy+ process of simulating a model starts by converting the model from the **MDL** form to executable form in order to assure maximum performance. This is done using the **BUILDMODEL** command. Once the model has been converted to executable form, then the **SIMULATE** command uses the executable form for its simulations. Models only need to be converted to executable form once! As long as the model stays the same, the executable can be reused many times with different simulation scenarios. This two-step process allows the simulation of the model to execute at machine speed.

The BUILDMODEL command

The **BUILDMODEL** command works by converting the model's **MDL** to an equivalent Fortran model. For the **BUILDMODEL** command to function properly, there must be a Fortran compiler installed on the machine and Modeleasy+ must be told where to write the model's Fortran source. After the location for the source is set by the **LIBRARY** command, the **BUILDMODEL** command is used to convert the model.

```
:_library buildmodel "c:\source"
:_buildmodel klein1
*Linkule BUILDMOD. Version 12/5/98*
Default value of 24 used for MAXOBS_.
Number of equations : 6
Number of endogenous variables: 6
Ordering results:
  0 prerecursive equations;
  5 simultaneous equations;
  1 postrecursive equations;
  1 feedback variables.
The Mortran source of the model is in the file:
  c:\source\KLEIN1.MOR
```

In addition to simply converting the model's **MDL** to an equivalent Fortran model, the **BUILDMODEL** command also analyzes the model and reorders the equations of the model into pre-recursive, simultaneous and post-recursive blocks. This reordering of equations also greatly enhances the simulation speed since the number of iterations is significantly reduced.

The SIMULATE command

Having converted the model to executable form, the **SIMULATE** command is used to simulate the model. Before doing the simulation, the time horizon must be set by defining an object called **TSRANGE** which has the desired dates. Any other simulation options, such as iteration limits, should also be set before doing the simulation. This is done for the **Klein1** model below.

```
:_tsrange = 1923 1 1935 1
:_simoption iterlimit 100
*Linkule SIMOPTION. Version 5/1/98*
:_simulate klein1 : print
*Linkule SIMULATE. Version 8/1/98*
*Subroutine T4SIMS - Version 31.aug.98 12:00*
"DYNAMIC" simulation
TSRANGE: 1923 1 TO 1935 1
Actual number of simulation periods = 13 on a maximum of 24
PERIOD 1 REQUIRED 17 ITERATIONS
PERIOD 2 REQUIRED 24 ITERATIONS
PERIOD 3 REQUIRED 23 ITERATIONS
PERIOD 4 REQUIRED 30 ITERATIONS
PERIOD 5 REQUIRED 29 ITERATIONS
PERIOD 6 REQUIRED 31 ITERATIONS
PERIOD 7 REQUIRED 26 ITERATIONS
PERIOD 8 REQUIRED 22 ITERATIONS
PERIOD 9 REQUIRED 30 ITERATIONS
PERIOD 10 REQUIRED 29 ITERATIONS
PERIOD 11 REQUIRED 27 ITERATIONS
PERIOD 12 REQUIRED 28 ITERATIONS
PERIOD 13 REQUIRED 26 ITERATIONS
```

As can be seen, the number of iterations required for convergence is reported for each simulated time period.

The simulated values of the variables have the same name as the variable with an "S" placed in front of the name. As can be seen below, the simulated values for C are named SC.

```
:_c
 C (A Time Series with 22 Components)
 39.8 41.9 45 49.2 50.6 52.6 55.1 56.2 57.3 57.8 55 50.9 45.6
 46.5 48.7 51.3 57.7 58.7 57.5 61.6 65 69.7
:_sc
 SC (A Time Series with 13 Components)
 50.336 55.695 56.705 51.342 46.021 46.954 52.646 54.935 54.735
 51.815 50.489 51.93 53.317
```

There are better methods for examining the simulated values. These are described in the following section.

Examining Results

The results of a simulation or other analysis, are most easily examined using time series display capabilities or graphical display capabilities.

Continuing with the simulation of the **Klein1** model, the values of C and the simulated values, SC, can be nicely tabulated with the **TSTABULATE** command as seen below, where the percent difference is also displayed.

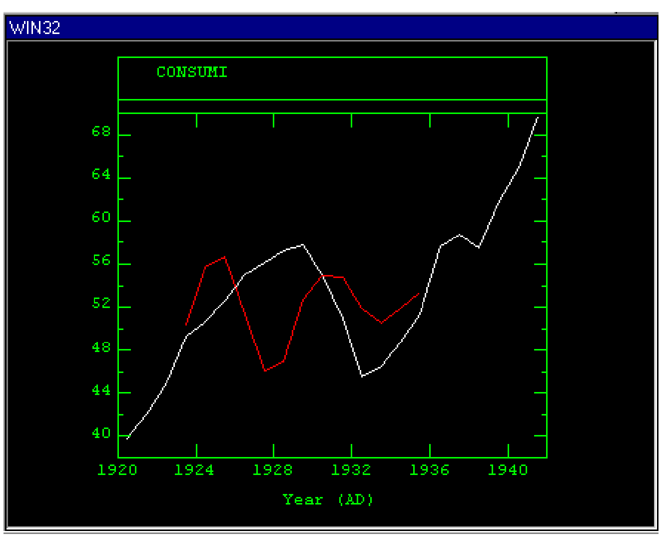
```
:_pctdiff = (sc-c)/c
:_tstabulate c sc pctdiff

      YEAR    C      SC      PCTDIFF
      ****   ****   ******   *********
      1923   49.2   50.336    .023095
      1924   50.6   55.695    .10069
      1925   52.6   56.705    .07804
      1926   55.1   51.342   -.068211
      1927   56.2   46.021   -.18112
      1928   57.3   46.954   -.18057
      1929   57.8   52.646   -.08917
      1930   55     54.935   -.0011826
      1931   50.9   54.735    .075347
      1932   45.6   51.815    .1363
      1933   46.5   50.489    .08578
      1934   48.7   51.93     .066319
      1935   51.3   53.317    .039317
```

Gross features of the data can be easily seen from the table, e.g., there is a dip in the data for C which starts in 1930 and ends in 1935 with the minimum being in 1932. The simulated result, SC, shows a similar dip, but that starts in 1926 and ends in 1930 with a minimum in 1927.

While the results displayed in tables are useful for many purposes, trends are far easier to examine using visual tools such as graphs produced by the graphical capabilities of Modeleasy+. In order to draw a graph, it is necessary to initialize the device where the graph will be drawn (**WIN32** for the Windows display in the following), do any tailoring of the display required, and then graph the data.

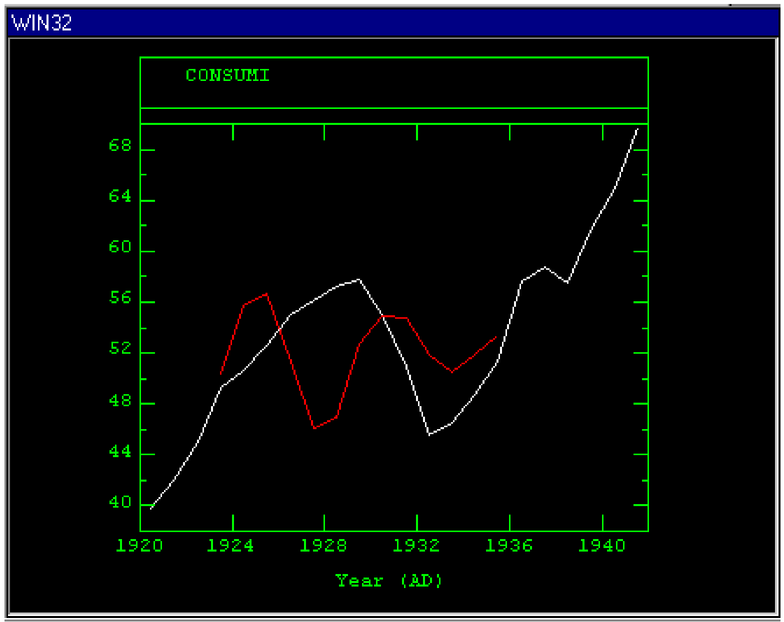
```
:_graphics win32
:_setxaxis useraxis numbered:1
:_xtmvalues = 1920, 1924, 1928, 1932, 1936, 1940, 1944
:_xtmlabels = xtmvalues
:_setxlabel "Year (AD)"
:_tsgraph c
:_setcolor red
:_tsaddgraph sc
```



From this graph, the trends mentioned above can be more easily seen than from the tabulated results. The dip in the consumption data, C, plotted in white can easily be seen, as well as that of the simulation results, SC, plotted in red.

Better results can be obtained by including add factors or constant adjustment terms in the model. This is done by using the CA keyword with the **ESTIMATE** and **BUILDMODEL** commands. There is no need to explicitly change the model, the CA keyword automatically adjusts the model for these terms. The series of commands below demonstrate the re-analysis of the **Klein1** including the add factors terms. This analysis is the same as that presented previously, the only difference being the use of the CA keyword on the **ESTIMATE** and **BUILDMODEL** command lines.

```
:_getdeck klein1 on models
:_getlist klein1v on data
:_outputoption noprint
*Linkule OUTPUTOP. Version 5/1/98*
:_estimate klein1 : ca
*Linkule ESTIMATE. Version 5/1/98*
 The solution for equation C has been derived.
 The solution for equation I has been derived.
 The solution for equation W1 has been derived.
:_buildmodel klein1 : ca
*Linkule BUILDMOD. Version 12/5/98*
Default value of 24 used for MAXOBS_.
Number of equations : 6
Number of endogenous variables: 6
Ordering results:
  0 prerecursive equations;
  5 simultaneous equations;
  1 postrecursive equations;
  1 feedback variables.
The Mortran source of the model is in the file:
  c:\source\KLEIN1.MOR
:_tsrange = 1923 1 1935 1
:_simoption iterlimit 100
*Linkule SIMOPTION. Version 5/1/98*
:_simulate klein1
*Linkule SIMULATE. Version 8/1/98*
*Subroutine T4SIMS - Version 31.aug.98 12:00*
:_graphics win32
:_setxaxis useraxis numbered:1
:_xtmvalues = 1920, 1924, 1928, 1932, 1936, 1940, 1944
:_xtmlabels = xtmvalues
:_setxlabel "Year (AD)"
:_tsgraph c
:_setcolor red
:_tsaddgraph sc
```



As can be easily seen from the graph above, including add factors in the analysis of the **Klein1** model brings the simulated values into complete agreement with the historical values.