User Manual: PrioSim

Contents

Installation (x64)

Unzip Simulator.zip folder anywhere. To run the Simulator change to the "Program" directory with

cd ".\Program".

Make a parameter file (see below), containing variables, expressions and parameters to the program Run the program from Stata with the command "Simulator <path to parameter file>". The results from the simulation are stored as .CSV and can be opened I Excel or be imported back to Stata for further analysis.

NB: Make sure that the Parameter variables in the file match the data in Stata.

Configuration

To configure the program create a text file and add parameters to it. There are several types of parameters. The basic syntax is TYPE; COLUMNNAME; EXPRESSION.

The PrioSim Stata plug in can currently handle Multinomial logit regressions. The plug in reads the parameter file, interpret it, runs the regression and passes data and parameters to the application. Here is a sample parameter file:

```
//Parameters for estimation command and simulation

UNITID; statenum

TIMEID; year

DEP; sipcat

//When using nv() function it has to be calculated before use

SUPPVAR; timeinstatus; if(lv(sipcat)==nv(sipcat)|lv(timeinstatus) + 1|0)

SUPPVAR; timesinceindependance; lv(timesinceindependance)+1

SUPPVAR; lnpop; lv(lnpop)

SUPPVAR; growthrate; umean(growthrate|lt)+(0.95*(lv(growthrate)-
umean(growthrate|lt)))

IDEP; sip1 lagged; if(nv(sipcat)==1.0|1|0)

IDEP; sip2 lagged; if(nv(sipcat)==2.0|1|0)

IDEP; sip3 lagged; if(nv(sipcat)==3.0|1|0)

IDEP; lngdpc lagged; lv(lngdpc)+nv(growthrate)

IDEP; pchange; 2.0^(-nv(timeinstatus)/4.0)
```

```
IDEP; pindependence; 2.0^(-nv(timesinceindependance)/8.0)

IDEP; dominantsharesip0 lagged; if(edgecountf(sipcat|0)/edgecount(0) >
0.50|1|0)

IDEP; dominantsharesip1 lagged; if(edgecountf(sipcat|1)/edgecount(0) >
0.50|1|0)

IDEP; dominantsharesip3 lagged; if(edgecountf(sipcat|3)/edgecount(0) >
0.50|1|0)

LINKS;neighbors

IFS; if year > 1950

ENDPARAMS; baseoutcome(0)

BETADRAWS; 10

SPLIT; none

SIMULATIONS; 25

ITERATIONS; 15

STARTTIME; 2000

RESULTFILE; U:\Joachim\PrioSim\ResultsTest.txt

LOGFILE; Logtest.txt
```

Parameter Types

Type	Explanation
UNITID	This is the name of the Stata column holding the id for the unit. Example of a units is a country. The column is expected to contain an integer id.
TIMEID	This is the name of the Stata column holding the id for the time. Example of a time is a year. The column is expected to contain an integer id. The program supposes that the combination of UNITKEY and TIMEKEY are unique.
DEP	This is the name of the Stata column holding values for the dependant variable.
IDEP	This is the name of the Stata column holding values for the independent variable. (The program expects an expression)
SUPPVAR	This is the name of the Stata column holding values for the

	supporting variables. (The program expects an expression)
LINKS	This is the name of the Stata column holding values for the unit edges. This should be unit ids separated by blanks. The program creates a graph based on this which can be used in graph calculations.
IFS	Stata syntax if command.
ENDPARAMS	Stata syntax commands. (End of mlogit command)
BETADRAWS	This is a constant which determines how many draws from the estimates that should be executed.
SIMULATIONS	This is a constant which determines how many simulations that should be executed for each draw.
ITERATIONS	This is a constant which determines how many time units that should be simulated for each simulation.
STARTTIME	The time when simulation start.
RESULTFILE	Path to result file.
LOGFILE	Filename. The file is stored in the "Log" folder. To have a log from the Stata use the regular Stata syntax.

Parameter Expressions

The program uses muParser math expression parser to evaluate expressions. See http://muparser.sourceforge.net/.

The following table gives an overview of the functions supported by the default implementation. It lists the function names, the number of arguments and a brief description.

Name	Argc.	Explanation
sin	1	sine function
cos	1	cosine function
tan	1	tangens function
asin	1	arcus sine function
acos	1	arcus cosine function
atan	1	arcus tangens function
sinh	1	hyperbolic sine function
cosh	1	hyperbolic cosine
tanh	1	hyperbolic tangens function
asinh	1	hyperbolic arcus sine

		function
acosh	1	hyperbolic arcus tangens function
atanh	1	hyperbolic arcur tangens function
log2	1	logarithm to the base 2
log10	1	logarithm to the base 10
log	1	logarithm to the base 10
ln	1	logarithm to base e (2.71828...)
exp	1	e raised to the power of x
sqrt	1	square root of a value
sign	1	sign function -1 if x<0; 1 if x>0
rint	1	round to nearest integer
abs	1	absolute value
if	3	if ... then ... else ...
min	var.	min of all arguments
max	var.	max of all arguments
sum	var.	sum of all arguments
avg	var.	mean value of all arguments

The following table lists the default binary operators supported by the parser.

Operator	Meaning	Priority
=	assignement*	-1
and	logical and	1
or	logical or	1
xor	logical xor	1
<=	less or equal	2
>=	greater or equal	2
!=	not equal	2
==	equal	2
>	greater than	2
<	less than	2
+	addition	3
-	subtraction	3
*	multiplication	4
/	division	4
^	raise x to the power of y	5

A set of custom functions is defined in the program to accommodate common simulation scenarios and calculation on the unit graph.

Name	Arg.	Explanation
lv	varname	Get the value for t-1.
nv	varname	Get the value for t. Use only for a variable that has been evaluated, meaning that it is above in the parameter file.
usum	varname, log, time	Get the sum for all units t-1.
umean	varname, time	Get the mean for all units t-1.
usumf	varname, filtervarname, filtervalue, log, time	Get the sum for all units t-1. With filter.
edgeno	0	Gets count of edges for unit.
edgenof	filtervarname, filtervalue	Gets count of edges for unit. With filter.
lt	0	Gets time t-1
t	0	Gets time t
lag	varname, time	Gets simulated value for variable t-time (if not simulated observed value is returned)
lagh	varname, time	Gets observed value for variable t-time.