SAS: Interfacing PROC MIANALYZE with PROC PANEL

As shown in Figure 1, the dataset raw_input contains a cross-sectional variable “country”, a time variable “year”, a response “Y”, and two covariates named “X1”, “X2”.

VIEWTABLE: Weblib.Raw_input

	Country	Year	X1	X2	Y
1	Algeria	2001	0.0673317564	0.4351944573	0.448066046
2	Algeria	2002	.	0.615258652	0.5529112399
3	Algeria	2003	0.9204722381	.	0.6209422684
4	Algeria	2004	.	0.7169796264	0.3961471951
5	Algeria	2005	0.6660961321	0.1110945472	0.6106332768
6	Armenia	2001	0.4572514073	.	0.7935189164
7	Armenia	2002	0.9891285127	0.0751261211	0.3596208182
8	Armenia	2003	.	0.6644486379	0.0832925833
9	Armenia	2004	0.7160875435	0.3631859021	0.9066725988
10	Armenia	2005	0.2022114792	.	0.8737188498
11	Canada	2001	0.7704879533	0.3114024186	0.7865263797
12	Canada	2002	0.8959476738	.	0.7085372554
13	Canada	2003	0.3028514241	0.7677049787	0.7944306574
14	Canada	2004	.	0.0634820813	0.3636754633
15	Canada	2005	0.2476976901	0.6632414699	0.2439505241

Figure 1. Raw panel dataset

Since covariates contain missing values, we have to impute them first by applying MCMC in PROC MI:

```
proc mi data = raw_input seed = 42037921 nimpute = 100
        out = imputed_data;
 mcmc timeplot(mean(x1) mean(x2));
 var x1 x2;

run;
```

which gives us 100 imputed datasets contained in a single dataset imputed_data (Figure 2).

VIEWTABLE: Multiply-Imputed Data

	Imputation Number	Country	Year	X1	X2	Y
13	1	Canada	2003	0.3028514241	0.7677049787	0.7944306574
14	1	Canada	2004	0.7661903293	0.0634820813	0.3636754633
15	1	Canada	2005	0.2476976901	0.6632414699	0.2439505241
16	2	Algeria	2001	0.0673317564	0.4351944573	0.448066046
17	2	Algeria	2002	0.2867451579	0.615258652	0.5529112399
18	2	Algeria	2003	0.9204722381	0.2171776982	0.6209422684
19	2	Algeria	2004	-0.026196493	0.7169796264	0.3961471951
20	2	Algeria	2005	0.6660961321	0.1110945472	0.6106332768
21	2	Armenia	2001	0.4572514073	0.8047710122	0.7935189164
22	2	Armenia	2002	0.9891285127	0.0751261211	0.3596208182
23	2	Armenia	2003	0.4567493619	0.6644486379	0.0832925833
24	2	Armenia	2004	0.7160875435	0.3631859021	0.9066725988
25	2	Armenia	2005	0.2022114792	0.670361454	0.8737188498
26	2	Canada	2001	0.7704879533	0.3114024186	0.7865263797
27	2	Canada	2002	0.8959476738	0.1958394508	0.7085372554
28	2	Canada	2003	0.3028514241	0.7677049787	0.7944306574
29	2	Canada	2004	1.0153749364	0.0634820813	0.3636754633
30	2	Canada	2005	0.2476976901	0.6632414699	0.2439505241
31	3	Algeria	2001	0.0673317564	0.4351944573	0.448066046
32	3	Algeria	2002	0.22668593	0.615258652	0.5529112399

Figure 2. Dataset imputed_data obtained from PROC MI.

Our final goal is to regress Y on X1, X2, and the first lag of Y. The latter is created through the following code:

```
proc sort data = imputed_data;
     by _Imputation_ country year;
run;

proc panel data = imputed_data;
     by _Imputation_;
     id country year;
     clag Y(1) / out = imputed_data;
run;
```

First, we fit a model in PROC REG, because MIANALYZE can be interfaced seamlessly with REG as follows:

```
proc reg data = imputed_data outest = out_reg covout noprint;
                model Y = Y_1 X1 X2;
                by _Imputation_;
run;
quit;

proc mianalyze data = out_reg;
     modeleffects Intercept Y_1 X1 X2;
run;
```

We need to do a similar thing with PROC PANEL. One of the reasons is that PANEL can do GMM estimation which is more appropriate than OLS when a lagged response is present in the right-hand side. We have to make the following code work:

```
proc panel data = imputed_data outest = out_pan covout noprint;
     by _Imputation_;
     id country year;
     inst  constant depvar exogenous = (X1 X2);
     model Y = Y_1 X1 X2 / itgmm nolevels;
run;


/* Here some code will be inserted   (A) */

proc mianalyze data = out_pan;
     modeleffects Intercept Y_1 X1 X2;
run;
```

Unfortunately, this code fails without some additional statements in line (A). While the structure of out_pan is similar to that of out_reg (Figure 3), PROC MIANALYZE cannot take out_pan “as is”.

	Imputation Number	Label of model	Type of statistics	Variable names for rows of estimated COV	Dependent variable	Root mean squared error	Intercept	Y_1	X1	X2
1	1	MODEL1	PARMS		Y	0.2586323481	0.2044861357	0.0554335382	0.3501548737	0.3751480328
2	1	MODEL1	COV	Intercept	Y	0.2586323481	0.1196785365	-0.054640924	-0.066136828	-0.118569162
3	1	MODEL1	COV	Y_1	Y	0.2586323481	-0.054640924	0.1068706785	-0.013067699	0.000340318
4	1	MODEL1	COV	X1	Y	0.2586323481	-0.066136828	-0.013067699	0.0766459234	0.0833686012
5	1	MODEL1	COV	X2	Y	0.2586323481	-0.118569162	0.000340318	0.0833686012	0.1771117671
6	2	MODEL1	PARMS		Y	0.2701682461	0.226614165	0.1089918822	0.2619299633	0.3186969873
7	2	MODEL1	COV	Intercept	Y	0.2701682461	0.184437032	-0.062021296	-0.12788967	-0.171665
8	2	MODEL1	COV	Y_1	Y	0.2701682461	-0.062021296	0.1131167449	-0.006562147	0.0027757458
9	2	MODEL1	COV	X1	Y	0.2701682461	-0.12788967	-0.006562147	0.1320244841	0.138162387
10	2	MODEL1	COV	X2	Y	0.2701682461	-0.171665	0.0027757458	0.138162387	0.2172807075
11	3	MODEL1	PARMS		Y	0.2722834599	0.7171481339	0.0701056313	-0.13188221	-0.294688196
12	3	MODEL1	COV	Intercept	Y	0.2722834599	0.2064273078	-0.079792887	-0.137467656	-0.206125099
13	3	MODEL1	COV	Y_1	Y	0.2722834599	-0.079792887	0.1220840173	-0.000229137	0.0271560235
14	3	MODEL1	COV	X1	Y	0.2722834599	-0.137467656	-0.000229137	0.1378445912	0.1578536805
15	3	MODEL1	COV	X2	Y	0.2722834599	-0.206125099	0.0271560235	0.1578536805	0.2660090898

	Imputation Number	Label of Model Statement	Estimation Method Used	Type of Observation	Name of Variable	Dependent Variable	Mean Square Error of Transformed Model	Cross Section ID	Intercept	Y_1	X1
1	1		_GMMITD_	ITGMM		Y	0.4472729117		0.1277761729	1.5281287653	**********
2	1		_GMMITD_	COVB	Intercept	Y	.		0.0105006533	0.0344633782	**********
3	1		_GMMITD_	COVB	Y_1	Y	.		0.0344633782	40.428720692	**********
4	1		_GMMITD_	COVB	X1	Y	.		0.0238169832	-12.1207865	**********
5	1		_GMMITD_	COVB	X2	Y	.		0	0	0
6	2		_GMMITD_	ITGMM		Y	0.1859706089		0.0630060106	0.9493438504	**********
7	2		_GMMITD_	COVB	Intercept	Y	.		0.0248156301	0.2662207879	**********
8	2		_GMMITD_	COVB	Y_1	Y	.		0.2662207879	3.5385330065	**********
9	2		_GMMITD_	COVB	X1	Y	.		-0.046367987	-0.636145273	**********
10	2		_GMMITD_	COVB	X2	Y	.		0	0	0
11	3		_GMMITD_	ITGMM		Y	0.3964842248		0.1085945245	1.5706780832	**********
12	3		_GMMITD_	COVB	Intercept	Y	.		0.0510685064	0.7363252868	**********
13	3		_GMMITD_	COVB	Y_1	Y	.		0.7363252868	11.994639548	**********
14	3		_GMMITD_	COVB	X1	Y	.		-0.182445425	-3.00397961	**********
15	3		_GMMITD_	COVB	X2	Y	.		0	0	0
16	4		_GMMITD_	ITGMM		Y	0.6491175954		0.0484808617	0	**********
17	4		_GMMITD_	COVB	Intercept	Y	.		0.036934968	0	-0.1361039
18	4		_GMMITD_	COVB	Y_1	Y	.		0	0	0
19	4		GMMITD	COVB	X1	Y	.		-0.1361039	0	**********

Figure 3. Output of PROC REG, out_reg (top) and output of PROC PANEL, out_pan (bottom).

For instance, the output variable _TYPE_ is supposed to tell apart point estimates of regression coefficients from their variance-covariance matrix. For point estimates, _TYPE_ should be “PARM” or “PARMS”, but in out_pan it is equal to “ITGMM”. Another distinction is that PANEL outputs MSE as opposed root MSE in REG.

Taking information from out_pan and converting it manually into out_reg’s format seems like a way to go. The catch is that it will fail unless you also assign correct TYPE of the dataset (do not confuse a dataset attribute TYPE with the variable _TYPE_ mentioned above). The TYPE attribute of out_reg is equal to EST. If you create a copy of out_reg as:

```
data out_reg2;
     set out_reg;
run;
```

then, unexpectedly, PROC MIANALYZE won't accept out_reg2:

```
47   proc mianalyze data = out_reg2;
48        modeleffects Intercept Y_1 X1 X2;
49   run;

ERROR: The input TYPE= data set is not a valid data set without
specifying variables for standard errors in the STDERR statement.
```

because the TYPE attribute is not copied from out_reg to out_reg2. After this is taken into account, out_pan becomes good enough for MIANALYZE and the final code looks as follows:

```
proc panel data = imputed_data outest = out_pan covout noprint;
     by _Imputation_;
     id country year;
     inst  constant depvar exogenous = (X1 X2);
     model Y = Y_1 X1 X2 / itgmm nolevels;
run;

                          /* Code (A) begins */

/* Compute _RMSE_ : */
data out_pan;
     set out_pan;
     retain _RMSE_;
     if (_MSE_ ne .) then _RMSE_ = sqrt(_MSE_);
     output;
run;

/* Re-assign some values and drop redundant columns: */
data out_pan;
     set out_pan;
     if _TYPE_ eq 'ITGMM' then _TYPE_ = 'PARMS';
     if _TYPE_ eq 'COVB' then _TYPE_ = 'COV';
     _MODEL_ = 'MODEL1';
     drop _METHOD_  _CSID_   _MSE_;
run;
```

```
/* Change the order of variables */
data out_pan;
     RETAIN _Imputation_ _MODEL_ _TYPE_
            _NAME_ _DEPVAR_ _RMSE_
            Intercept
            Y_1 X1 X2 Y;
     set out_pan;
run;

/* Set outpan's type to EST */
proc datasets;
     modify out_pan (TYPE = EST);
quit;

                        /* Code (A) ends */

proc mianalyze data = out_pan;
     modeleffects Intercept Y_1 X1 X2;
run;
```