BBVA Research

XTSELVAR & XTSELMOD: Selection of Variables and Specification in a Panel Data Framework

Virtual Conference Stata, USA Meeting
July 30-31, 2020

Alfonso Ugarte-Ruiz

Creating Opportunities

Contents

Motivation

- Evaluating the forecasting/prediction accuracy of a statistical model is becoming increasingly common and essential in a broad range of practical applications (e.g. macroeconomics variables forecasting for regulatory purposes, machine-learning and big-data techniques, etc.)

- In the 2019 Spanish Stata Conference we presented various new commands that allow evaluating the out-of-sample prediction performance of panel-data models in their time-series and cross-individual dimensions separately (***xtoos_t and xtoos_i***). (see Stata Conference Madrid 2019 or https://ideas.repec.org/c/boc/bocode/s458710.html)

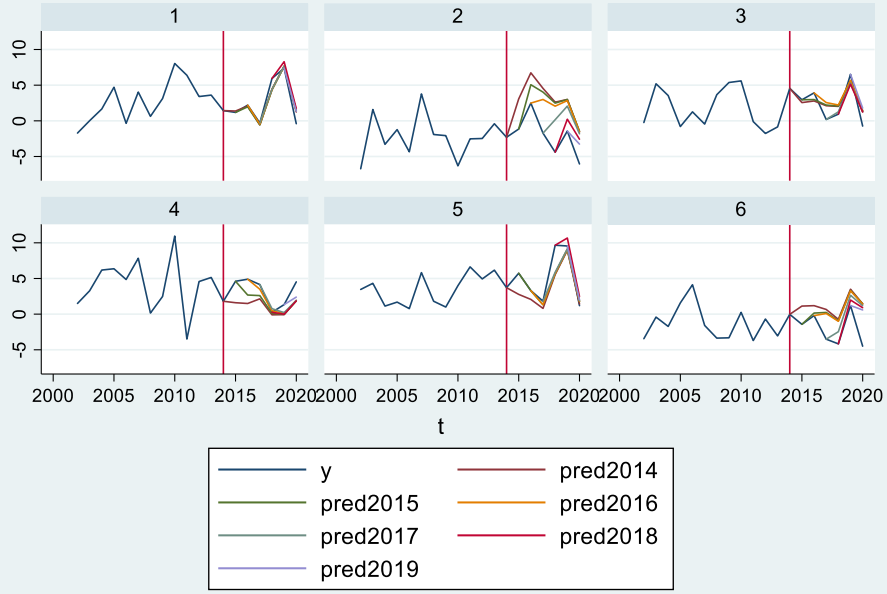
- ***xtoos_t*** and ***xtoos_i*** were based on the idea that evaluating the prediction performance of a panel-data model should take into account the two dimensions inherent in a panel, the time-series dimension and the cross-section (individuals) dimension.

- **Now we have built upon those commands to use prediction accuracy as a metric to rank and select across different sets of variables and specifications in a panel data framework, (commands *xtselvar* and *xtselmod*)**

- These new commands could be installed through the package xtsel:

  ssc install xtsel

  (https://ideas.repec.org/c/boc/bocode/s458816.html)



xtoos_t excludes a number of time periods for each individual in the panel.

Then for the remaining subsample it fits the specified model and uses the resulting parameters to forecast the dependent variable in the unused periods (out-of-sample).

xtoos_i excludes a group of individuals (e.g. countries) from the estimation sample (including all their observations throughout time).

Then for each remaining subsample it fits the specified model and uses the resulting parameters to predict the dependent variable in the unused individuals (out-of-sample).

- Some previously available procedures in Stata that perform **cross-validation** exercises (e.g. *crossfold, cvauroc*) usually play with all the observations when separating the in- and out-of-samples, without taking into account if such observations could belong to different individuals or are subsequent observations from the same individual.

- The latter could be problematic if, for instance, one wants to fit a dynamic or a Fixed-Effects model, or could simply make the results more difficult to analyze in a panel data framework

- There are also other similar existing Stata procedures that allow computing all possible models fitted by a command to a dependent variable from a set of predictors, like *allpossible* and *tuples*.

- The new commands **xtselvar** and **xtselmod** allow us to perform a similar exercise to "*allpossible*" but allowing to evaluate and rank different predictors and specifications using both traditional in-sample statistics and also out-of-sample prediction performance, while allowing several options that are usually required or useful in a panel data framework.

Common features of the new procedures

- *xtselvar* helps us to select the best predictor between a number of alternative explanatory variables (candidates). The procedure estimates the same defined specification *n* times, keeping constant the same dependent variable and an optional list of fixed control variables.

- *xtselmod* helps us to select the best specification between all possible combinations of a defined set of explanatory variables. It relies on the command *tuples*. Given n possible explanatory variables, the procedure estimates 2^n - 1 different specifications, one per each possible combination.

- For each candidate variable/specification, the procedure estimates a set of parameters and statistical criteria.

    1. Adjusted R squared, R2_ad
    2. Akaike Information Criterion, AIC
    3. Bayesian Information Criterion, BIC
    4. U-Theil in time-series dimension: RMSE of variable/specification vs. RMSE from a naïve prediction or an AR1 model, Uth_TS
    5. U-Theil in cross-section dimension: RMSE of variable/specification vs. RMSE from a naïve prediction or an AR1 model, Uth_CS

- Both commands rank each variable/specification according to each criterion and generates one ranking per each one of them.

- They also compute a composite ranking summarizing all five criteria. They finally sort all candidate variables/specifications according to the selected ranking, which by default is the composite ranking.

- *xtselvar* also reports coefficients and t-statistic of each candidate variable

- Both commands allow choosing weights for each one of the five criteria used to compute the composite ranking. They also allow ranking the variables/specifications according to a selected criterion of preference.

- For instance, if the primary objective of the estimation is to obtain the most accurate prediction of the dependent variable, the user could choose to rank the specifications according **only** to their forecasting ability, i.e. according to the estimated U-Theil in its time-series dimension.

- They allow choosing different estimation methods including some dynamic methodologies and could also be used in a dataset with only time-series observations.

- In the case the specification includes lags of the dependent variable, the procedure automatically generates dynamic forecasts for the out-of-sample evaluation performance.

- In the case of the out-of-sample evaluation in the time-series dimension, they allow choosing an exact horizon ***h*** at which to evaluate the forecasting performance of the model including the candidate variable.

- It also allows us to estimate the forecasting performance from horizon ***t+1*** until ***t+h***.

- ***xtselvar*** and ***xtselmod*** require packages ***matsort***, ***tuples*** and ***xtoos*** to be installed

- Both procedures' options and characteristics also allow us the following:

1. To specify a list of variables that will remain fixed in the specification.
2. To display the results of each estimation for each variable/specification or just show a final summary with each variable/specification ordered according to the score in the final ranking
3. To create a log file that saves each variable results and the final summary
4. To create an excel file to save the final summary

- The procedures' options and characteristics share most of the same options than ***xtoos_t*** and ***xtoos_i***:

1. Choosing different estimation methods
2. Choosing dynamic methods (**xtabond/xtdpdsys**)
3. Choosing between a naïve prediction or an AR1 model as the alternative/comparison model
4. Choosing the estimation method of the AR1 model
5. Using dynamic specifications (lags of the dependent variable). They automatically handle **dynamic forecasting**
6. Could be used automatically in a dataset with only time-series observations
7. Using data with different time frequencies, i.e. annual, quarterly, monthly and undefined time-periods
8. Evaluating the model's performance of one particular individual or a defined group of individuals instead of the whole panel
9. Choosing between within (FE), random (RE) or dummy variables estimation
10. To include, or not, the estimated individual component (intercept) in the prediction

- *xtselvar* and *xtselmod* require packages *matsort*, *tuples* and *xtoos* to be installed.

- Paul Millar, 2005. "MATSORT: Stata module to sort a matrix by a given column,"Statistical Software Components S449504, Boston College Department of Economics, revised 28 Jan 2009.
- Joseph N. Luchman & Daniel Klein & Nicholas J. Cox, 2006. "TUPLES: Stata module for selecting all possible tuples from a list",Statistical Software Components S456797, Boston College Department of Economics, revised 17 May 2020.
- Alfonso Ugarte-Ruiz, 2019. "XTOOS: Stata module for evaluating the out-of-sample prediction performance of panel-data models,"Statistical Software Components S458710, Boston College Department of Economics, revised 09 Jun 2020.

xtselvar:
Selection of variables
from within different groups

- *xtselvar* saves and presents the results of the analysis in different ways. The user can choose to display the results of each estimation for each variable and it can also create a log file to save all the results or an excel file to save the final summary

- The procedure displays a final summary through a table that shows all the statistics estimated for each candidate variable, the ranking according to each criterion, and the composite ranking. The table of results is displayed ordered by the criterion selected by the user

- The syntax of the command is as following:

Syntax

```
xtselvar depvar [indepvars] [if] [in], *indate(string) *cdate(string) *ksmpl(integer)
        [fixed(varlist)] [met(string)] [mcomp(string)] [evalopt(varname)]
        [fe] [xbu] [dum] [opar] [lags(numlist)] [qui] [log(string)] [exc(string)] [she(string)]
        [ord(string)] [down] [weights(numlist)] [hor(integer)] [uphhor] [groups(integer)] [ncomp(integer)]
        [pca1(varlist)] [pca2(varlist)] ... [pca10(varlist)]
        [model_options]
```

- Use of ***xtselvar*** to classify 21 different variables, x1 and z1_1, z1_2,...z1_20. The dates at which the time-series out-of-sample evaluation starts and end must be specified, the same as the number of individuals left-out at each partition in the cross-section out-of-sample evaluation

```
. sysuse panelexample, clear
. xtset id t
```

```
. xtselvar y x1 z1_1-z1_10 z1_11-z1_20, indate(2015) cdate(2020) ksmpl(100)
```

	coef	tstat	R2	AIC	BIC	Uth_TS	Uth_CS	R2_r	AIC_r	BIC_r	Uth_TS_r	Uth_CS_r	Total
x1	.5721471	35.35003	.0751199	49806.92	49821.34	.8172297	.9613279	1	1	1	1	1	5
z1_7	-.6353409	-32.96722	.0661446	49903.5	49917.92	.8223203	.9659972	2	2	2	2	2	10
z1_10	-.6168185	-31.49129	.0614955	49953.16	49967.58	.8224611	.9683829	3	3	3	3	3	15
z1_8	.6801396	31.61562	.0610574	49957.82	49972.25	.8241922	.9686356	4	4	4	4	4	20
z1_1	.8578814	27.75835	.050166	50073.15	50087.57	.8286728	.9740842	5	5	5	5	5	25
z1_4	-.8774976	-27.15462	.0466003	50110.62	50125.04	.8302152	.975943	6	6	6	7	6	31
z1_5	.8533368	25.12984	.0387825	50192.29	50206.71	.8295736	.9800185	7	7	7	6	7	34
z1_2	.873935	24.69856	.0376018	50204.56	50218.98	.8335772	.980502	8	8	8	9	8	41
z1_9	.8619397	24.64965	.0375093	50205.52	50219.95	.8313972	.9806218	9	9	9	8	9	44
z1_6	.4206233	8.417895	.0041779	50545.97	50560.39	.8448027	.9973912	10	10	10	10	10	50
z1_3	.1477366	2.884084	.0002367	50585.47	50599.89	.8486403	.9992786	11	11	11	11	11	55
z1_20	-.0124964	-.4851169	.0001625	50586.21	50600.63	.8490108	.9995058	12	12	12	14	14	64
z1_11	-.0322123	-1.225816	.0000135	50587.7	50602.12	.8491204	.9994405	13	13	13	19	12	70
z1_17	.0338847	1.318586	-.0000988	50588.82	50603.24	.8488619	.9995462	14	14	14	12	16	70
z1_13	.0117738	.4561327	-.0001955	50589.79	50604.21	.8490068	.9994764	17	17	17	13	13	77
z1_15	-.0106642	-.4114143	-.0001922	50589.76	50604.18	.8490446	.9995651	15	15	15	16	17	78
z1_16	.0049144	.1886311	-.0001932	50589.77	50604.19	.8490444	.9995739	16	16	16	15	18	81
z1_14	-.0042419	-.1626071	-.0001995	50589.83	50604.25	.8490626	.9996013	19	19	19	17	20	94
z1_12	.0179272	.6901911	-.0001967	50589.8	50604.22	.8493196	.9996534	18	18	18	21	21	96
z1_19	-.0087565	-.3364492	-.0001996	50589.83	50604.25	.849064	.9995776	20	20	20	18	19	97
z1_18	-.0048976	-.1869624	-.0001996	50589.83	50604.25	.8491448	.9995187	21	21	21	20	15	98

- If we want to always include in the specification the variables x2, x3, x4 and x5, we should used the option fixed():

```
. xtselvar y x1 z1_1-z1_10 z1_11-z1_20, indate(2015) cdate(2020) ksmpl(100) fixed(x2 x3 x4 x5) qui
```

	coef	tstat	R2	AIC	BIC	Uth_TS	Uth_CS	R2_r	AIC_r	BIC_r	Uth_TS_r	Uth_CS_r	Total
x1	.6055874	88.65824	.6487739	40616.49	40659.75	.5125209	.592126	1	1	1	1	1	5
z1_7	-.6740465	-77.27456	.6369776	40946.73	40989.99	.5185894	.6019815	2	2	2	2	2	10
z1_1	-.7762325	-71.39568	.6285991	41175.16	41218.42	.5271409	.6088805	3	3	3	3	3	15
z1_6	.8248707	67.2554	.623654	41307.63	41350.9	.5284274	.6130063	4	4	4	4	4	20
z1_2	.8174138	68.77339	.6232277	41319.32	41362.58	.5314136	.6132126	5	5	5	5	5	25
z1_10	-.9159055	-57.1896	.6114957	41625.58	41668.85	.5387273	.6226445	6	6	6	6	6	30
z1_8	.8994465	50.97655	.6026412	41851.22	41894.48	.5422656	.6297824	7	7	7	7	7	35
z1_4	.901108	48.13759	.599123	41939.5	41982.76	.5489541	.6325888	8	8	8	8	8	40
z1_3	.457238	17.26714	.5683072	42678.93	42722.2	.5644402	.6564048	9	9	9	9	9	45
z1_5	-.2888129	-10.65531	.5638276	42783.34	42826.6	.5702432	.6597798	10	10	10	10	10	50
z1_9	-.0950386	-3.461858	.5618384	42828.72	42871.98	.5715701	.6613432	11	11	11	11	11	55
z1_16	-.0216158	-1.555403	.5615909	42834.45	42877.72	.5716834	.6615249	12	13	13	12	13	63
z1_15	.0155391	1.115417	.5615848	42834.17	42877.43	.5717661	.6614935	13	12	12	15	12	64
z1_12	-.0072212	-.5247058	.5615388	42835.29	42878.56	.571748	.6616333	14	14	14	13	21	76
z1_11	.0132066	.9547499	.5615384	42835.74	42879	.571895	.6615322	15	15	15	20	14	79
z1_18	.0015965	.115751	.561515	42836.18	42879.44	.571781	.6615627	16	16	16	17	16	81
z1_14	.0027065	.1982075	.5615077	42836.24	42879.5	.5717906	.6615389	19	18	18	18	15	88
z1_17	.006295	.4588308	.5615147	42836.29	42879.56	.5717656	.6615845	17	20	20	14	19	90
z1_20	-.0150219	-1.080867	.5614646	42836.2	42879.46	.5719592	.6615741	21	17	17	21	18	94
z1_19	-.0264281	-1.913761	.5614752	42836.25	42879.51	.5718411	.6615723	20	19	19	19	17	94
z1_13	.0003639	.0262478	.5615122	42836.33	42879.6	.5717688	.6616201	18	21	21	16	20	96

- If we want to show each variable results and saving them in a log file named "results", we should use the option log():

```
. xtselvar y x1 z1_1-z1_20, indate(2015) cdate(2020) ksmpl(100) log(results)
```

- If we do not want to show each variable results, and we want to save the final summary table in an excel file named "results" and the worksheet named "results1", we should use the options qui and exc() together with the option she(). Options exc() and she() must be used together:

```
. xtselvar y x1 z1_1-z1_20, indate(2015) cdate(2020) ksmpl(100) exc(results) she(results1)
```

- If we want to give null weights to the adjusted R2, AIC and BIC, and equal weights to the U-Theil in time-series and cross-section dimensions, we should use the option ***weights()***. The given weights should be between 0 and 1:

```
. xtselvar y x1 z1_1-z1_10 z1_11-z1_20, indate(2015) cdate(2020) ksmpl(100) wei(0 0 0 0.5 0.5)
```

- If we want to order the final summary according to the R-squared in a descending order, we should use the options *ord()* and *down*:

```
. xtselvar y x1 z1_1-z1_10 z1_11-z1_20, indate(2015) cdate(2020) ksmpl(100) ord(R2_ad) down
```

- If we want to specify an exact horizon at which the time-series out-of-sample performance should be evaluated, we should use the option *hor()*:

```
. xtselvar y x1 z1_1-z1_10 z1_11-z1_20, indate(2015) cdate(2020) ksmpl(100) hor(3)
```

- If instead of an exact horizon, we want to evaluate the out-of-sample performance between horizons 1 and 3, we should used options *hor()* and *uph* together.

```
. xtselvar y x1 z1_1-z1_10 z1_11-z1_20, indate(2015) cdate(2020) ksmpl(100) hor(3) uph
```

- Use of PCA to construct composite control variables

- *xtselvar* allows generating a number of principal components (through PCA) for one or more groups of variables (topics) so that these components can be used as fixed control variables in each regression.

- This option could be specially useful in the case that there is a too large number of possible control variables that cannot be included altogether in each regression.

- Given that testing all possible combinations might be unfeasible, we can group them into a smaller set of principal components that act as uncorrelated control variables.

- It could also be useful to perform an initial selection of variables when all the predictors could be classified within smaller groups of similar/related variables.

- We could be able to select the best predictors from each group, while using as control variables principal components from the rest of groups.

- This strategy might help us to avoid the bias from omitting the control variables from all other groups different than the group in which the selection is being made.

- If we want to create thre principal components from three groups of variables with 20 variables each, e.g. groups z2 and z3: z2_1, z2_2 ... z2_20 and z3_1, z3_2 ... z3_20, we should use the options *groups()* and options *pca#()*, in this case *pca1()* … *pca3()*.

- The option *groups()* defines how many groups of variables are and thus how many principal components should be estimated and included in the specification. The options *pca1()*... to *pca#groups()* should list the variables within each group. There should be as many lists as groups of variables and therefore the number of *groups()* and the number of lists should coincide.

```
. xtselvar y x4 z4*, indate(2015) cdate(2020) ksmpl(100) groups(3) pca1(z1*) pca2(z2*) pca3(z3*)
```

	coef	tstat	R2	AIC	BIC	Uth_TS	Uth_CS	R2_r	AIC_r	BIC_r	Uth_TS_r	Uth_CS_r	Total
x4	1.097217	96.82433	.618369	41446.36	41489.62	.5286907	.6173411	1	1	1	1	1	5
z4_2	-1.135135	-57.40556	.5543977	42996.56	43039.82	.5699594	.6669674	2	2	2	2	2	10
z4_4	-1.122296	-56.13097	.5511318	43069.41	43112.67	.5743789	.6693668	3	3	3	3	3	15
z4_5	-1.119598	-51.13135	.5435234	43237.63	43280.89	.5768461	.6749488	4	4	4	4	4	20
z4_6	1.12223	49.1081	.5352151	43417.19	43460.45	.5879245	.6811095	5	5	5	5	5	25
z4_7	-1.078287	-44.17836	.5316783	43493.02	43536.28	.5879592	.6838044	6	6	6	6	6	30
z4_1	.9254395	33.45347	.5114818	43916.07	43959.34	.5994477	.6981991	7	7	7	7	7	35
z4_3	.7155696	23.37116	.4966775	44214.82	44258.08	.6119791	.7086366	8	8	8	8	8	40
z4_9	.3397432	10.50667	.4860146	44424.57	44467.83	.615176	.716266	9	9	9	10	9	46
z4_8	-.3474656	-10.73166	.4854331	44435.54	44478.8	.6143874	.7165848	10	10	10	9	10	49
z4_11	.0348125	2.090383	.48279	44487.2	44530.46	.6166877	.7184221	11	11	11	11	11	55
z4_19	.0269305	1.628275	.4827689	44487.46	44530.72	.6167643	.7184845	12	12	12	12	14	62
z4_20	.0153269	.9263316	.4827188	44488.2	44531.46	.6170883	.7184416	13	13	13	21	12	72
z4_13	.0093939	.5697255	.4826951	44488.37	44531.63	.6168385	.7185267	16	14	14	13	18	75
z4_10	.0266262	.8001289	.4827032	44488.56	44531.83	.6168898	.7185732	14	15	15	17	21	82
z4_14	-.0176045	-1.07812	.4826994	44488.91	44532.17	.6169434	.718522	15	16	16	20	16	83
z4_12	-.0025144	-.1554491	.4826597	44489.23	44532.49	.6168988	.7184693	18	17	17	19	13	84
z4_15	.0098343	.5955335	.4826617	44489.67	44532.93	.6168847	.7185237	17	20	20	15	17	89
z4_16	-.0019788	-.1199373	.4826554	44489.57	44532.83	.6168943	.7185104	19	19	19	18	15	90
z4_17	.0039925	.2421044	.4826354	44489.33	44532.59	.6168885	.7185289	21	18	18	16	19	92
z4_18	-.0120663	-.7444793	.4826439	44489.86	44533.12	.6168389	.7185367	20	21	21	14	20	96

- We can also generate various principal components from just one large group of variables, for instance if we do not have an a priori classifications of the predictors. We can, for example create 6 components from all variables whose name starts with z, using also the option ncomp().

- Additionally, we should specify only one group in option groups() and list all variables z* in the option pca1():

```
. selectvar y x4 z4* if sample==1, ind(2014) cd(2020) o(100) k(100) r(0) qui groups(1) pca1(z*) ncomp(6)
```

	coef	tstat	R2	AIC	BIC	Uth_TS	Uth_CS	R2_r	AIC_r	BIC_r	Uth_TS_r	Uth_CS_r	Total
x4	1.097425	97.01525	.6186391	41439.32	41482.59	.5283757	.6171234	1	1	1	1	1	5
z4_2	-1.135236	-57.46271	.5545421	42993.34	43036.6	.5697188	.6668615	2	2	2	2	2	10
z4_4	-1.122196	-56.17284	.5512527	43066.75	43110.01	.5742258	.6692776	3	3	3	3	3	15
z4_5	-1.120545	-51.23457	.5437483	43232.72	43275.98	.5766032	.6747823	4	4	4	4	4	20
z4_6	1.123388	49.2221	.5354589	43412	43455.26	.5877162	.6809237	5	5	5	5	5	25
z4_7	-1.078582	-44.23006	.5318393	43489.6	43532.86	.5878272	.6836869	6	6	6	6	6	30
z4_1	.9263655	33.51755	.5116549	43912.56	43955.83	.5993298	.6980737	7	7	7	7	7	35
z4_3	.7149043	23.36423	.4967522	44213.39	44256.65	.6119135	.7085783	8	8	8	8	8	40
z4_9	.3415537	10.57078	.4861338	44422.29	44465.55	.6150484	.7161784	9	9	9	10	9	46
z4_8	-.3484874	-10.77125	.4855374	44433.56	44476.82	.6143018	.7165089	10	10	10	9	10	49
z4_11	.033909	2.037492	.4828658	44485.77	44529.03	.6166165	.7183638	11	11	11	11	11	55
z4_19	.0272273	1.647316	.4828538	44485.85	44529.11	.616683	.7184233	12	12	12	12	14	62
z4_20	.0149202	.9023699	.4827995	44486.67	44529.93	.6170044	.7183783	13	13	13	21	12	72
z4_13	.0097086	.589208	.4827799	44486.76	44530.03	.6167584	.7184579	15	14	14	13	16	72
z4_10	.029404	.8841187	.4827939	44486.84	44530.11	.6168033	.7185002	14	15	15	15	21	80
z4_12	-.0028795	-.1781397	.4827441	44487.63	44530.89	.6168213	.7184038	18	17	17	19	13	84
z4_14	-.016968	-1.039865	.4827798	44487.39	44530.65	.6168644	.718458	16	16	16	20	17	85
z4_16	-.0023359	-.141676	.4827393	44487.99	44531.25	.6168175	.7184472	19	19	19	18	15	90
z4_15	.0104875	.6355216	.4827462	44488.07	44531.33	.6168058	.7184596	17	20	20	16	18	91
z4_17	.0039839	.2417477	.4827185	44487.78	44531.04	.616813	.7184675	21	18	18	17	19	93
z4_18	-.0107487	-.6636266	.4827261	44488.31	44531.57	.6167721	.7184754	20	21	21	14	20	96

xtselmod:
Selection/ranking of specifications

- *xtselmod* saves and presents the results of the analysis in different ways. The user can choose to display the results of each estimation for each specification and it can also create a log file to save all the results, or an excel file to save the final summary.

- The procedure displays a final summary through a table that shows all the five statistics estimated for each candidate specification, the ranking of each specification according to each criterion, and the composite ranking. The table of results is displayed ordered by the criterion selected by the user

- The syntax of the command is as the following:

Syntax

```
xtselmod depvar [indepvars] [if] [in], *indate(string) *cdate(string) *ksmpl(integer)
        [conditionals(string)] [fixed(varlist)] [met(string)] [mcomp(string)] [evalopt(varname)]
        [fe] [xbu] [dum] [opar] [lags(numlist)] [qui] [log(string)] [exc(string)] [she(string)]
        [ord(string)] [down] [weights(numlist)] [hor(integer)] [uphhor]
        [spec1(varlist)] [spec2(varlist)] ... [spec10(varlist)]
        [model_options]
```

- Use of ***xtselmod*** to classify specifications based on variables x1, x2, x3, x4 and x5 (32 models) The dates at which the time-series out-of-sample evaluation starts and end must be specified, the same as the number of individuals left-out at each partition in the cross-section out-of-sample evaluation

```
. xtselmod y x1 x2 x3 x4 x5, indate(2015) cdate(2020) ksmpl(100) qui
```

	Model	R2_ad	AIC	BIC	Uth_TS	Uth_CS	R2_ad_r	AIC_r	BIC_r	Uth_TS_r	Uth_CS_r	_Total_
1.	x1 x2 x3 x4 x5	.6487739	40616.49	40659.75	.5125209	.592126	1	1	1	1	1	5
2.	x1 x2 x3 x4	.6251073	41268.48	41304.53	.5245405	.6117508	2	2	2	2	2	10
3.	x1 x2 x4 x5	.6069672	41726.07	41762.12	.5433187	.6267006	3	3	3	3	3	15
4.	x1 x3 x4 x5	.5824692	42344.98	42381.03	.5633461	.645786	4	4	4	4	4	20
5.	x2 x3 x4 x5	.5615563	42834.35	42870.4	.5717682	.6615247	5	5	5	5	5	25
6.	x1 x4 x5	.557319	42926.64	42955.48	.5807574	.6652162	6	6	6	6	6	30
7.	x2 x3 x4	.5385966	43344.53	43373.37	.5813962	.6786239	7	7	7	7	7	35
8.	x1 x2 x3 x5	.5354852	43412.09	43448.14	.5905009	.6806926	8	8	8	8	8	40
9.	x1 x3 x4	.5295043	43539.31	43568.15	.5912008	.6855711	9	9	9	9	9	45
10.	x2 x4 x5	.5188397	43750.68	43779.52	.6018768	.6932898	10	10	10	11	10	51
11.	x1 x2 x3	.5129666	43885.05	43913.89	.6002825	.6969981	11	11	11	10	11	54
12.	x3 x4 x5	.4942322	44261.95	44290.79	.6171581	.7106149	12	12	12	12	12	60
13.	x1 x2 x5	.4918172	44298.62	44327.46	.6184268	.7123159	13	13	13	13	13	65
14.	x4 x5	.4684642	44755.12	44776.75	.634882	.7287092	14	14	14	15	14	71
15.	x2 x3 x5	.4475722	45144.77	45173.61	.6433206	.7423654	15	15	15	17	15	77
16.	x1 x2 x4	.4454019	45183.24	45212.08	.6345546	.7442215	16	16	16	14	16	78
17.	x3 x4	.4421133	45242.58	45264.21	.6408539	.7463441	17	17	17	16	17	84
18.	x2 x3	.4257495	45531.73	45553.36	.6512808	.7568426	18	18	18	18	18	90
19.	x2 x5	.402983	45910.3	45931.93	.6712236	.7720574	19	19	19	19	19	95
20.	x2 x4	.3582354	46642.89	46664.53	.685638	.8005532	20	20	20	20	20	100
21.	x1 x3 x5	.3515532	46747.2	46776.04	.6973602	.8045281	21	21	22	22	21	107
22.	x1 x4	.3510796	46753.91	46775.55	.6910567	.8052925	22	22	21	21	22	108
23.	x1 x5	.3356961	46988.35	47009.98	.7069536	.8144889	23	23	23	24	23	116
24.	x1 x2	.3309844	47058.49	47080.13	.6992551	.8171602	24	24	24	23	24	119
25.	x1 x3	.2711573	47915.32	47936.96	.7307349	.8530526	25	25	25	25	25	125
26.	x4	.2630096	48025.86	48040.28	.7371215	.8580357	26	26	26	26	26	130
27.	x3 x5	.26138	48048.68	48070.3	.7419957	.8585769	27	27	27	27	27	135

- If we want to keep some variables fixed in the specification, we should use the option *fixed(),* for instance variable x5

```
. xtselmod y x1 x2 x3 x4, indate(2015) cdate(2020) ksmpl(100) fixed(x5) qui
```

	Model	R2_ad	AIC	BIC	Uth_TS	Uth_CS	R2_ad_r	AIC_r	BIC_r	Uth_TS_r	Uth_CS_r	_Total_
1.	x1 x2 x3 x4	.6487739	40616.49	40659.75	.5125209	.592126	1	1	1	1	1	5
2.	x1 x2 x4	.6069672	41726.07	41762.12	.5433187	.6267006	2	2	2	2	2	10
3.	x1 x3 x4	.5824692	42344.98	42381.03	.5633461	.645786	3	3	3	3	3	15
4.	x2 x3 x4	.5615563	42834.35	42870.4	.5717682	.6615247	4	4	4	4	4	20
5.	x1 x4	.557319	42926.64	42955.48	.5807574	.6652162	5	5	5	5	5	25
6.	x1 x2 x3	.5354852	43412.09	43448.14	.5905009	.6806926	6	6	6	6	6	30
7.	x2 x4	.5188397	43750.68	43779.52	.6018768	.6932898	7	7	7	7	7	35
8.	x3 x4	.4942322	44261.95	44290.79	.6171581	.7106149	8	8	8	8	8	40
9.	x1 x2	.4918172	44298.62	44327.46	.6184268	.7123159	9	9	9	9	9	45
10.	x4	.4684642	44755.12	44776.75	.634882	.7287092	10	10	10	10	10	50
11.	x2 x3	.4475722	45144.77	45173.61	.6433206	.7423654	11	11	11	11	11	55
12.	x2	.402983	45910.3	45931.93	.6712236	.7720574	12	12	12	12	12	60
13.	x1 x3	.3515532	46747.2	46776.04	.6973602	.8045281	13	13	13	13	13	65
14.	x1	.3356961	46988.35	47009.98	.7069536	.8144889	14	14	14	14	14	70
15.	x3	.26138	48048.68	48070.3	.7419957	.8585769	15	15	15	15	15	75

- Or we can obtain the same outcome by using the option *conditionals()* in the following way:

```
. xtselmod y x1 x2 x3 x4 x5, indate(2015) cdate(2020) ksmpl(100) conditionals(5) qui
```

- The option conditionals() also allows imposing more complicated restrictions, such as variables x1 and x2 should always go together:

```
. xtselmod y x1 x2 x3 x4 x5, indate(2015) cdate(2020) ksmpl(100) conditionals(!(1&!2) !(2&!1)) qui
```

	Model	R2_ad	AIC	BIC	Uth_TS	Uth_CS	R2_ad_r	AIC_r	BIC_r	Uth_TS_r	Uth_CS_r	_Total_
1.	x1 x2 x3 x4 x5	.6487739	40616.49	40659.75	.5125209	.592126	1	1	1	1	1	5
2.	x1 x2 x3 x4	.6251073	41268.48	41304.53	.5245405	.6117508	2	2	2	2	2	10
3.	x1 x2 x4 x5	.6069672	41726.07	41762.12	.5433187	.6267006	3	3	3	3	3	15
4.	x1 x2 x3 x5	.5354852	43412.09	43448.14	.5905009	.6806926	4	4	4	4	4	20
5.	x1 x2 x3	.5129666	43885.05	43913.89	.6002825	.6969981	5	5	5	5	5	25
6.	x3 x4 x5	.4942322	44261.95	44290.79	.6171581	.7106149	6	6	6	6	6	30
7.	x1 x2 x5	.4918172	44298.62	44327.46	.6184268	.7123159	7	7	7	7	7	35
8.	x4 x5	.4684642	44755.12	44776.75	.634882	.7287092	8	8	8	9	8	41
9.	x1 x2 x4	.4454019	45183.24	45212.08	.6345546	.7442215	9	9	9	8	9	44
10.	x3 x4	.4421133	45242.58	45264.21	.6408539	.7463441	10	10	10	10	10	50
11.	x1 x2	.3309844	47058.49	47080.13	.6992551	.8171602	11	11	11	11	11	55
12.	x4	.2630096	48025.86	48040.28	.7371215	.8580357	12	12	12	12	12	60
13.	x3 x5	.26138	48048.68	48070.3	.7419957	.8585769	13	13	13	13	13	65
14.	x5	.2450805	48266.22	48280.64	.7522367	.8681399	14	14	14	14	14	70
15.	x3	.1819156	49069.77	49084.19	.7715502	.9035599	15	15	15	15	15	75

- Comparing particular specifications

- *xtselmod* also allows comparing and ranking up to 10 particular specifications.

- This option could be useful when the user wants to compare some particular specifications that have restrictions that are difficult to handle through the option conditionals, for instance when they involve interactions, or lags of the same variable

- It could also be useful when only a handful of possible specifications are to be compared.

- This option does not make use of the command tuples and do not find a combination of a set of variables, it just directly compares and rank the literal specifications introduced by the user.

- If we want to compare, for instance, 3 particular specifications without combining them up, we should use options spec1() up to spec3().

- If we would want to compare ten specifications, which is the maximum in this type of options, we should use options spec1() up to spec10().

- Inside each one of the parenthesis we should write down each specification we want to try. Alternatively, we can only write down the part of each specification that is different from the other ones, and include in the option fixed() the common parts of the specification that remains constant in all the cases, for instance:

```
xtselmod y, indate(2015) cdate(2020) ksmpl(100) ///
spec1(x1 c.x1#c.x2 x3) spec2(c.x1#c.x2 x2 x3) spec3(x1 c.x1#c.x2 c.x2#c.x3) fix(x4 x5)
```

	Model	R2_ad	AIC	BIC	Uth_TS	Uth_CS	R2_ad_r	AIC_r	BIC_r	Uth_TS_r	Uth_CS_r	_Total_
1.	Specification 2	.6303777	41127.63	41170.89	.5254935	.6076701	1	1	1	1	1	5
2.	Specification 1	.5823916	42346.97	42390.23	.5633851	.6460065	2	2	2	2	2	10
3.	Specification 3	.5772232	42439.6	42482.86	.5649081	.6500933	3	3	3	3	3	15

Conclusions

Conclusions

- We have developed two new commands that allow testing and classifying the performance of different variables and specifications according to several in-sample and out-of-sample statistics.

- The main novelty of the commands is twofold:
    1. They help us to use the out-of-sample prediction performance as a selection criterion
    2. They are specially adapted for a panel data framework, firstly because the out-of-sample performance is measured in the two inherent dimensions of a panel, and secondly because they allow a large number of methodological options that typically are necessary in panel data analysis.

Another novel characteristic on one of the commands is that it allows generating a number of principal components (through PCA) for one or more groups of variables (topics) so that these components can be used as fixed control variables in each regression, a strategy that might help reducing the bias from omitting control variables