Empirical Model Discovery and Theory Evaluation

Code Supplement

Jurgen A. Doornik and David F. Hendry

University of Oxford

citation:
Doornik, J.A. and Hendry, D.F. (2014), "Empirical Model Discovery and Theory Evaluation: Code Supplement", University of Oxford.
Published online as supplement to
Hendry, D.F. and Doornik, J.A. (2014), "Empirical Model Discovery and Theory Evaluation: Automatic Selection Methods in Econometrics", The MIT Press.

version: 2014-July

Contents

Data and Software

This code supplement provides the data and code to replicate almost all results in the book. Chapter and section numbers refer to the original book. Sections specific to the supplement are unnumbered.

A. Required software

Most supplied code performs Monte Carlo experiments. The following *OxMetrics* software is required to run the supplied code:

PcGive The PcGive class for *Ox* is used by many experiments to implement both estimation and *Autometrics*-based model selection. A few experiments also require the PcNaive class that is part of PcGive. The version used is *PcGive* 14, see Hendry and Doornik (2013), part of the *OxMetrics* 7 family.
OxMetrics batch files are used for the empirical applications.

Ox *Ox* is required to run all Ox programs. The version used in this book is Ox Professional 7, see Doornik (2013), part of the *OxMetrics* 7 family. The PcGive class requires the professional version of *Ox*.

Autometrics Some experiments do not use *PcGive*, but instead use the Autometrics class together with PcFimlEx. The *Autometrics* version is 1.5g.

B. Discovery package installation

The package is provided in a zip file `discovery100.zip` (or a higher version number). It is recommended to create a `discovery` folder in your Ox installation:
`OxMetrics7/ox/packages/discovery`
and unzip the folder into there. However, other locations are also feasible.

C. Discovery package contents

- `discovery/book`
  The book folder contains the code to replicate most estimations and experiments in the Model Discovery book. The programs, described in the remainder of this document, are identified by the chapter and section number of the book.

- `discovery/code`
  The code folder holds all the code that implements the basic functionality of all experiments. The filenames of shared code start with `sim`:

  sim_1cut code specific for 1-cut simulations;

  sim_pcgive provides PcGiveExp to run an experiment using PcGive;

  sim_autometrics provides AutometricsExp to run an experiment using Autometrics for selection, and PcGive or PcFiml for estimation;

  simdesign base class for design of DGPs;

  simdesigns implement specific DGPs;

  simstore class to store simulation results and create a report;

  simutils generic helper functions.

  Further files in the code folder are

  autometrics header and compiled Ox code for Autometrics;

  PcFimlEx class that extends PcFiml for use with Autometrics.

- `discovery/data`
  The following three data sets are used in the book:

  dataz.in7/dataz.bn7 An extension of the original *PcGive* artificial data set.

  DHSY.xlsx *UK consumption*.
  This set of UK consumption data was collected quarterly, but not seasonally adjusted, for the period 1957:1 to 1976:2. It has been documented and analyzed by Davidson, Hendry, Srba, and Yeo (1978).

  HooverPerez(1999).xlsx Data set originally created for the Monte Carlo experiments of Hoover and Perez (1999).

  Three additional data sets are included:

  DHSY(updated).in7/DHSY(updated).bn7 *UK consumption*.
  An updated version of the DHSY data.

  TobinFoodUpdate14.in7/TobinFoodUpdate14.bn7 *US food expenditure data*, 1929–2007.
  This set of annual variables for the US was first analyzed by Tobin (1950), and previously investigated by a number of studies reported in Magnus and Morgan (1999), including Hendry (1999), based on the update of the time series in Tobin (1950) to 1989 by Magnus and Morgan (1999). It was extended to 2002 by Reade (2008), with results reported in Hendry (2009).

  UKMacro14.in7/UKMacro14.bn7
  UK annual macroeconomic data, 1870–2011.
  This set of annual macro variables for the UK has previously been analyzed by Ericsson, Hendry, and Prestwich (1998). It is an extension of the data analyzed by Friedman and Schwartz (1982).

- `discovery/doc`
  Documentation of the code in HTML, as well as this document. Start by loading `index.html` in your (modern) browser.

4 Empirical Modeling Illustrated

4.2 A simultaneous equations model

Requirements

Program	`04_02_dataz_sem.fl`
Data	`dataz.in7/dataz.bn7`
Software	PcGive
Output	`04_02_dataz_sem.out`

DataZ

The `dataz` data set contains the artificial DGP from PcGive, equations (4.1)–(4.4), together with 20 IIN[0, 1] variables:

`CONS`	c_t,
`INC`	i_t,
`INFLAT`	Δp_t,
`OUTPUT`	q_t,
`z0` ... `z19`	IIN[0, 1].

Results

The OxMetrics batch file replicates the simultaneous equations model that corresponds to the DGP:

1. loads the data,
2. transforms the data,
3. formulates the model,
4. estimates the model.

4.4 Modeling the artificial data consumption function

Requirements

Program	`04_04_dataz.fl`, `04_04_dataz_inflat.fl`
Data	`dataz.in7/dataz.bn7`
Software	PcGive
Output	`04_04_dataz.out`, `04_04_dataz_inflat.out`

Results

`04_04_dataz.fl` replicates the automatic model selection leading to (4.11). `04_04_dataz_inflat.fl` runs automatic model selection with IIS and SIS for Δp_t.

When `04_04_dataz.fl` estimates the model for `CONS`, *Autometrics* first reports the GUM, then performs the lag presearch. This is followed by the tree search:

```
[1.0] Start of Autometrics tree search
Searching from GUM 0  k=  13 loglik=    -221.244
Found new terminal 1  k=   5 loglik=    -230.906 SC=    3.1950
Found new terminal 2  k=   4 loglik=    -228.378 SC=    3.1295

Searching for contrasting terminals in terminal paths
Found new terminal 3  k=   6 loglik=    -229.327 SC=    3.2072

Encompassing test against GUM 0 removes: none

p-values in GUM 1 and saved terminal candidate model(s)   ** table removed ***

Searching from GUM 1  k=   8 loglik=    -226.724 LRF_GUM0( 5) [0.0733]
Recalling terminal 1  k=   5 loglik=    -230.906 SC=    3.1950
Recalling terminal 2  k=   4 loglik=    -228.378 SC=    3.1295
Recalling terminal 3  k=   6 loglik=    -229.327 SC=    3.2072

Searching for contrasting terminals in terminal paths
```

At the end, there are three terminal candidate models, of wich terminal 2 has the lowest BIC, and so is selected as the final model:

```
[2.0] Selection of final model from terminal candidates: terminal 2

p-values in Final GUM and terminal model(s)
                 Final GUM  terminal 1  terminal 2  terminal 3
CONS_1          0.00000000  0.00000000  0.00000000  0.00000000
INC             0.00000000  0.00000000  0.00000000  0.00000000
INC_1           0.00000000  0.00000000  0.00000000  0.00000000
INFLAT          0.44304521           .  0.00000000           .
INFLAT_1        0.13418790  0.00000000           .  0.00000000
OUTPUT          0.93129214           .           .  0.00955489
OUTPUT_1        0.44198008  0.00549272           .           .
S1973(3)        0.54933332           .           .  0.01863869
k                        8           5           4           6
parameters               9           6           5           7
loglik             -226.72     -230.91     -228.38     -229.33
AIC                 3.0614      3.0767      3.0309      3.0692
HQ                  3.1334      3.1248      3.0709      3.1253
SC                  3.2388      3.1950      3.1295      3.2072
                                           =======

coefficients and diagnostic p-values in Final GUM and terminal model(s)
                 Final GUM  terminal 1  terminal 2  terminal 3
CONS_1             0.80135     0.82056     0.81043     0.79213
INC                0.51002     0.50858     0.50950     0.51092
INC_1             -0.27998    -0.26680    -0.30006    -0.28322
INFLAT            -0.38018           .    -0.99583           .
INFLAT_1          -0.57333    -0.98355           .    -0.94878
OUTPUT          -0.0026339           .           .   -0.051835
OUTPUT_1         -0.037595   -0.056677           .           .
S1973(3)          -0.59627           .           .     -1.1025
k                        8           5           4           6
parameters               9           6           5           7
loglik             -226.72     -230.91     -228.38     -229.33
sigma               1.0870      1.1055      1.0839      1.0980
AR(5)              0.52185     0.46163     0.45079     0.47974
ARCH(4)            0.19351     0.02292     0.48769     0.12644
Normality          0.59135     0.80695     0.37895     0.52876
Hetero             0.58834     0.96022     0.31849     0.88634
Chow(70%)          0.15582     0.32155     0.12305     0.38161
                                           =======
```

The summary at the end records the Autometrics settings, and how many models where estimated:

```
Summary of Autometrics search
initial search space     2^27  final search space        2^8
no. estimated models      108  no. terminal models         3
test form                LR-F  target size        Small:0.01
outlier detection          no  presearch reduction      lags
backtesting              GUM0  tie-breaker                SC
diagnostics p-value      0.01  search effort        standard
time                     0.05  Autometrics version      1.5e
```

8 Selecting a Model in One Decision

8.4 Monte Carlo Simulation for $N = 1000$

1-cut DGP

$$\begin{aligned} y_t &= \beta_1 z_{1,t} + \cdots + \beta_{10} z_{10,t} + \epsilon_t, \quad & \epsilon_t &\sim \mathsf{IN}[0,1] \\ z_t &= (z_{1,t}, \ldots, z_{1000,t})', & z_t &\sim \mathsf{IN}_{1000}[\mathbf{0}, \boldsymbol{I}_{1000}]. \end{aligned}$$

DGP coefficients										
	$k = 1$	2	3	4	5	6	7	8	9	10
β_k	0.06	0.08	0.09	0.11	0.13	0.14	0.16	0.17	0.19	0.21

1-cut GUM

$$y_t = \gamma_0^F + \sum_{k=1}^{1000} \gamma_k z_{k,t} + u_t.$$

$T = 2000$, $M = 1000$, z_t not fixed (i.e. redrawn in every individual experiment). The intercept is forced into the model, as indicated by the superscript F.

Requirements

Program	`08_04_1cut.ox`
Software	Ox 7
Dependencies	PcGive class (OxMetrics 7), `sim_1cut.ox`
Also uses	`simutils.ox`
Output	`08_04_1cut.out`, `08_04_1cut.in7`, `08_04_1cut.bn7`
Running time	about 40 minutes

The running time is just an indication, as we sometimes run the code on a desktop, at other times on a notebook. We mostly use Ox Professional, avoiding parallel loops for the recorded timings. This Monte Carlo has large sample size and many variables, so benefits from multi-threading in the `olsc` function: Ox Console will take more than twice as long.

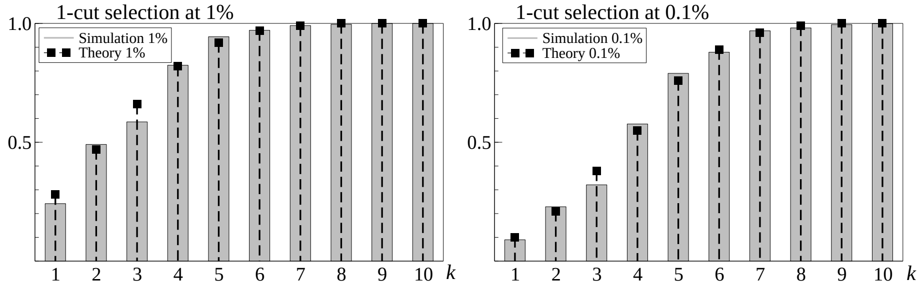

Figure 1
Retention rates $\widetilde{p}_k$ of relevant variables $z_k, k = 1, \ldots, 10$ using 1-cut rule with $\alpha = 1\%$ (left) and $\alpha = 0.1\%$ (right). $N = 1000, T = 2000, M = 1000$.

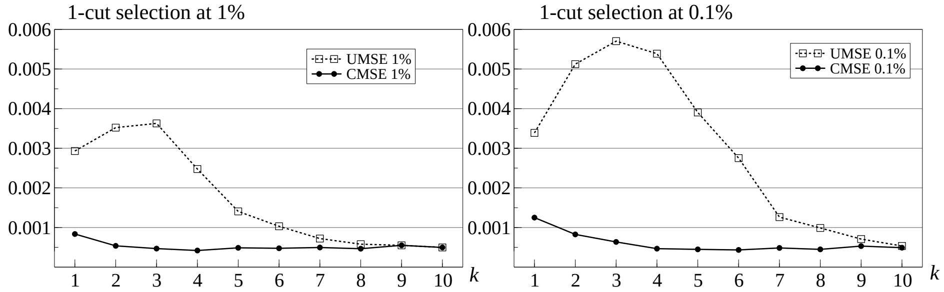

Figure 2
MSE of relevant variables $z_k, k = 1, \ldots, 10$ using 1-cut rule with $\alpha = 1\%$ (left) and $\alpha = 0.1\%$ (right). $N = 1000, T = 2000, M = 1000$.

1-cut results

α	Gauge	Potency
1%	1.00%	80%
0.1%	0.10%	68%

Table 1
Potency and gauge for 1-cut selection with 1000 variables.

8.5 Simulating MSE for $N = 1000$

Results generated in previous section; see Figure 2.

10 Bias Correcting Selection Effects

10.2 Bias correction after selection

Here we use a small version of the 1-cut experiment:

Small 1-cut DGP

$$\begin{aligned} y_t &= \beta_1 z_{1,t} + \beta_2 z_{2,t} + \beta_3 z_{3,t} + \epsilon_t, & \epsilon_t &\sim \mathsf{IN}[0,1] \\ \mathbf{z}_t &= (z_{1,t}, \ldots, z_{4,t})', & \mathbf{z}_t &\sim \mathsf{IN}_4[\mathbf{0}, \mathbf{I}_4]. \end{aligned}$$

DGP coefficients

	$k=1$	2	3
β_k	$4/T^{1/2}$	$2/T^{1/2}$	$1/T^{1/2}$

Small 1-cut GUM

$$y_t = \gamma_0^F + \sum_{i=1}^4 \gamma_i z_{i,t} + u_t.$$

So $z_{4,t}$ is an irrelevant variable, and the non-centralities are respectively: $4, 2, 1, 0$. $T = 100$, $M = 10^6$, $\mathbf{z}_t$ not fixed.

Requirements

Program	`10_02_1cut_biascorrection.ox`
Software	Ox 7
Dependencies	PcGive class, `sim_1cut.ox`
Also uses	`simutils.ox`
Output	`10_02_1cut_biascorrection.out`
Running time	about 10 minutes

Results

This program creates graphs, so is best run inside OxMetrics or using OxRun. See Figure 3 and Figure 4.

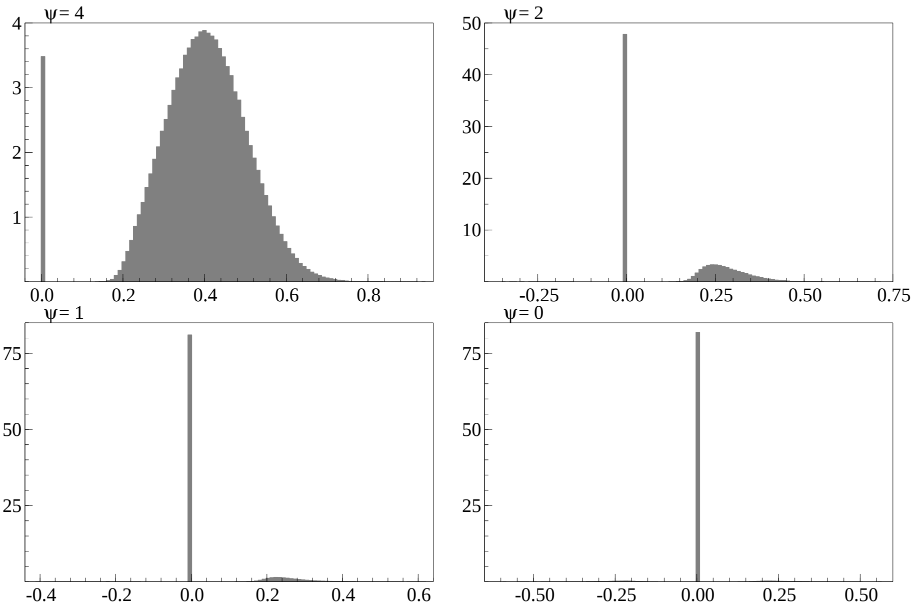

Figure 3
Unconditional distributions after 1-cut selection. Light: conditional distribution, dark: distribution after bias correction.

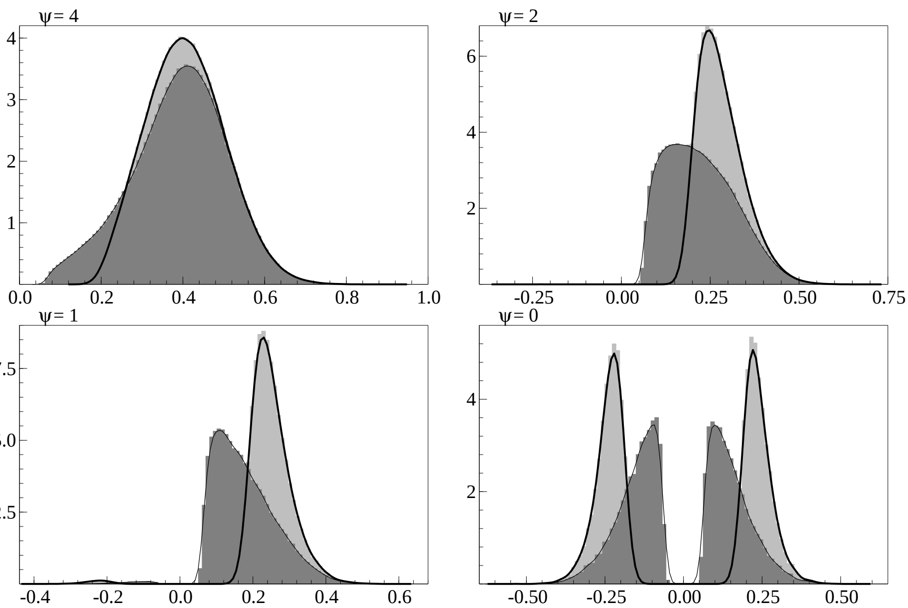

Figure 4
Impact of bias corrections on conditional distributions after 1-cut selection. Light: conditional distribution, dark: conditional distribution after bias correction.

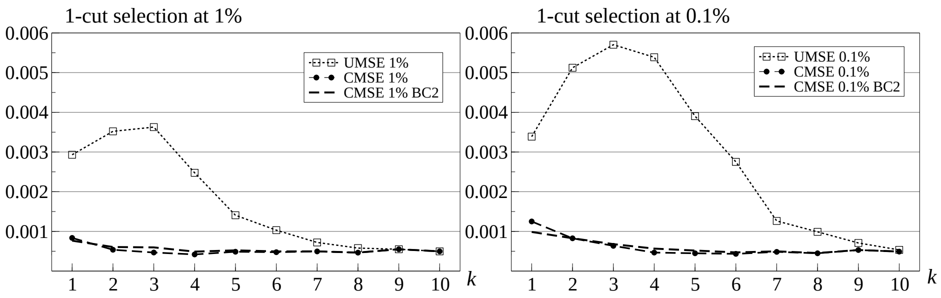

Figure 5
Impact of bias correction on CMSE_k for relevant variables at $\alpha = 1\%$ (left) and $\alpha = 0.1\%$ (right). BC2 refers to the two-step bias correction.

10.3 Impact of bias correction on MSE

Results generated in previous section.

Table 2
Effect of bias correction on CMSE and UMSE after 1-cut selection at 5%, $T = 100$, $M = 10^6$.

		Unconditional MSE after selection			MSE in GUM
	ψ	Uncorrected	BC 1-step	BC 2-step	
β_1	4	0.0143	0.0174	0.0193	0.0106
β_2	2	0.0257	0.0252	0.0258	0.0106
β_3	1	0.0128	0.0110	0.0105	0.0106
β_4	0	0.0029	0.0017	0.0013	0.0106
		Conditional MSE after selection			
	ψ	Uncorrected	BC 1-step	BC 2-step	
β_1	4	0.0093	0.0120	0.0139	
β_2	2	0.0108	0.0088	0.0100	
β_3	1	0.0270	0.0167	0.0132	
β_4	0	0.0587	0.0382	0.0282	

11 Comparisons of 1-cut Selection with *Autometrics*

11.5 Monte Carlo experiments for $N = 10$

The Castle et al. experiment can be implemented as a version of the 1-cut experiment:

Castle et al. DGP

$$
\begin{aligned}
y_t &= 5 + \textstyle\sum_{k=1}^{n} z_{k,t} + \epsilon_t, \quad \epsilon_t \sim \mathsf{IN}\left[0, \left(0.4n^{1/2}\right)^2\right] \\
\boldsymbol{z}_t &= (z_{1,t}, \ldots, z_{10,t})', \qquad \boldsymbol{z}_t \sim \mathsf{IN}_{10}\left[\boldsymbol{0}, \boldsymbol{I}_{10}\right].
\end{aligned}
$$

Castle et al. GUM

$$
y_t = \gamma_0^F + \textstyle\sum_{k=1}^{10} \gamma_k z_{k,t} + u_t.
$$

Using $n = 10$, this defines 10 experiments. $T = 75$, $M = 10^5$, $\boldsymbol{z}_t$ fixed (i.e. generated only once at the start, unlike chapter 10).

Requirements

Program	`11_05_1cut_cmp.ox`
Software	Ox 7
Dependencies	PcGive class, `sim_1cut.ox`
Also uses	`simutils.ox`
Output	`11_05_cmp1.out` (1-cut), `11_05_cmp2.out` (Autometrics, no diagnostics), `11_05_cmp3.out` (Autometrics), `11_05_cmp.in7/bn7`
Running time	from 5 to about 30 minutes

The `11_05_1cut_cmp.ox` program is run three times, with different settings.

11.6 Gauge and potency

Figure 6.

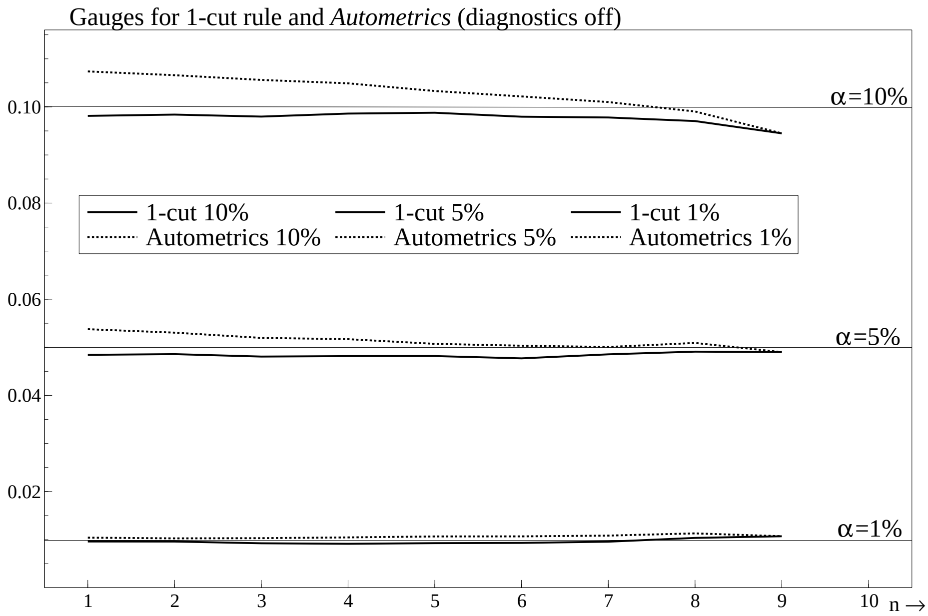

Figure 6
Gauges for 1-cut rule (solid lines) and *Autometrics* without diagnostic tracking (dotted lines) for $\alpha = 0.01, 0.05, 0.1$. The horizontal axis represents the $n = 1, \ldots, 10$ DGPs, each with n relevant variables. $T = 75, M = 10000$

11.7 Mean square errors

Figure 7.

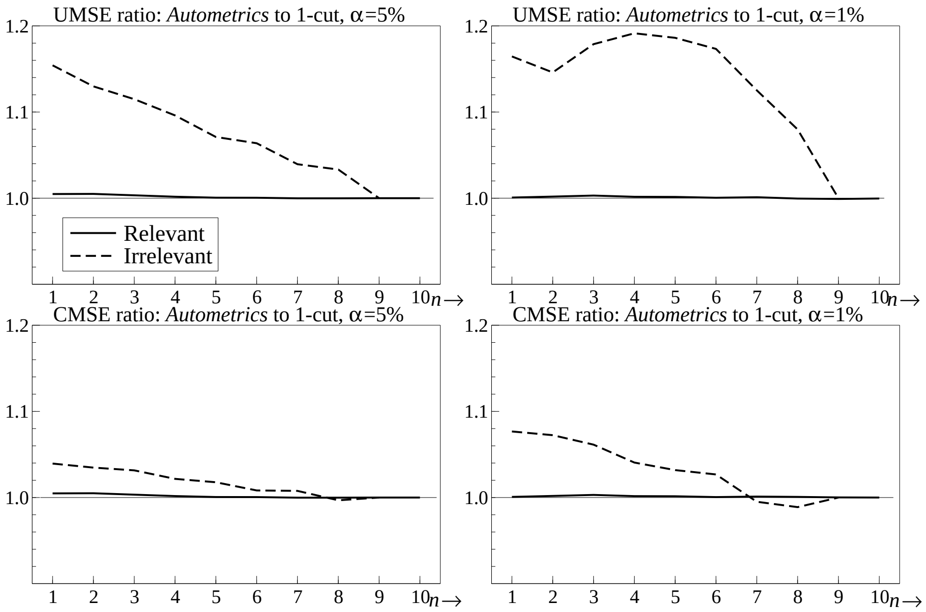

Figure 7
Ratios of MSEs for *Autometrics* to 1-cut rule as n changes

12 Impact of Diagnostic Tests

12.2 Selection effects on mis-specification tests

This section uses the JEDC experiment (see chapter 17) with independent regressors. The code for these experiments was first generated by PcNaive, and then adjusted as follows:

1. Add ARCH(1) test
   First define the constant:

```
enum
{
    TEST_ARCH = TEST_LAST
};
```

   Then add code to the predefined virtual functions:

```
CMyModel::GetTestName(const eval)
{   // return an array of strings with additional TEST_ names
    return {"ARCH(1)"};
}
CMyModel::GetTestIsTwoSided(const eval)
{   // return a row vector with a zero for each one-sided test,
    // and 1 for two-sided
    return <0>;
}
CMyModel::GetTest(const eval)
{   // return a 2 x c matrix with the TEST_ statistics in the
    // first row, and p-values in the second row
    return ARCHTest(1);
}
```

   Add the test:

```
    m_sys.AddEvalTest(TEST_ARCH     , 0, 0);
```

2. Adjust for AR(2) test:

```
    m_sys.AddEvalTest(TEST_AR        , 1, 2);
```

3. Set arguments for the ChowF test:

```
    m_sys.AddEvalTest(TEST_CHOWFORC   , 71, 1);
```

4. Store p-values of test

First add `m_mTestPvals` as a data member, then:

```
        m_mTestPvals ~= m_mTest[1][]';
```

5. Save p-values of test

First add `SaveTestPvals` as a function member, then:

```
    CPcNaiveExp::SaveTestPvals()
    {
        savemat(oxfilename(2) ~ "_p.in7", m_mTestPvals', m_asTest);
    }
```

Finally, call it:

```
        exp.SaveTestPvals();
```

Requirements

Program	`12_02_dgp.ox`, `12_02_gum.ox`, `12_02_autometrics.ox`
Program, large M	`12_02_dgp2.ox`, `12_02_gum2.ox`, `12_02_autometrics2.ox`
Software	Ox 7
Dependencies	PcNaive and PcGive class
Also uses	—
Output	`12_02_dgp.out`, `12_02_gum.out`, `12_02_autometrics.out`, `12_02_dgp2.out`, `12_02_gum2.out`, `12_02_autometrics2.out`, `*_d.in7/bn7` (data replication, not needed) `*_m.in7/bn7` (default Monte Carlo output, not needed) `*_p.in7/bn7` (p-values, used for plot)
Running time	from less than a minute to almost an hour

Results

The following tests are used as the default set for mis-specification testing in Autometrics:

Test	Distribution	In JEDC DGP
AR(p)	$F(p, T-k-p)$	$F(2, 93)$
ARCH(p)	$F(p, T-2p)$	$F(1, 98)$
Normality	$\chi^2(2)$	$\chi^2(2)$
Heteroscedasticity	$F(s, T-s-1)$	$F(10, 89)$
Chow	$F(T-\tau, \tau-k)$	$F(29, 66)$

Here T is the sample size of the regression, k the number of regressors (counting the intercept if included), s the number of regressors in the auxiliary regression (regressors and their non-redundant squares, excluding the intercept). The JEDC DGP (using $\rho = 0$: see section 17.6) has $T = 100$, no intercept and 5 regressors, so $k = 5$ and $s = 10$. The GUM has $k = 21$ and $s = 40$. The Chow test is at 70%, so for a break on or after observation $\tau = 71$.

See Figure 8, Figure 9 and Table 3.

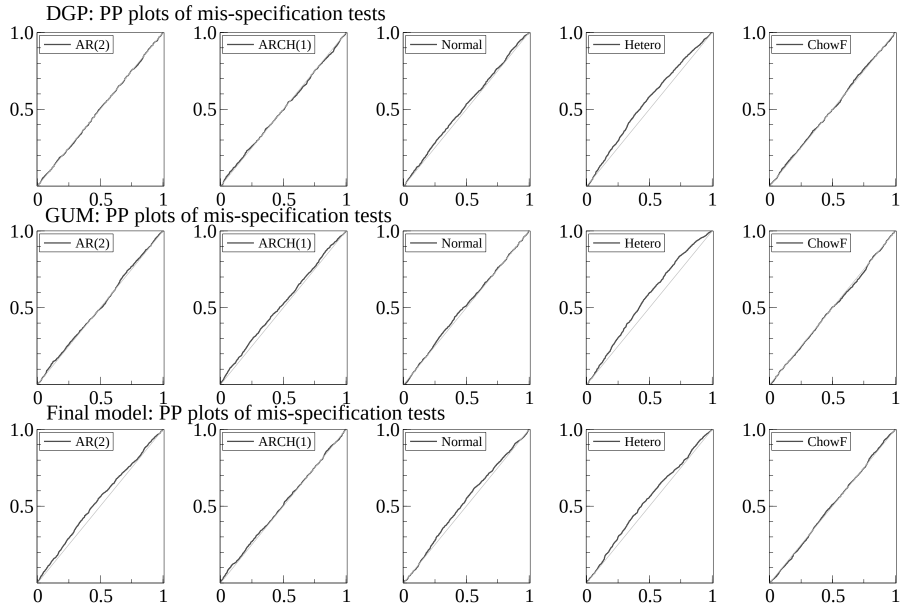

Figure 8
Calibrating mis-specification test null rejection frequencies, $T = 100, M = 1000$.

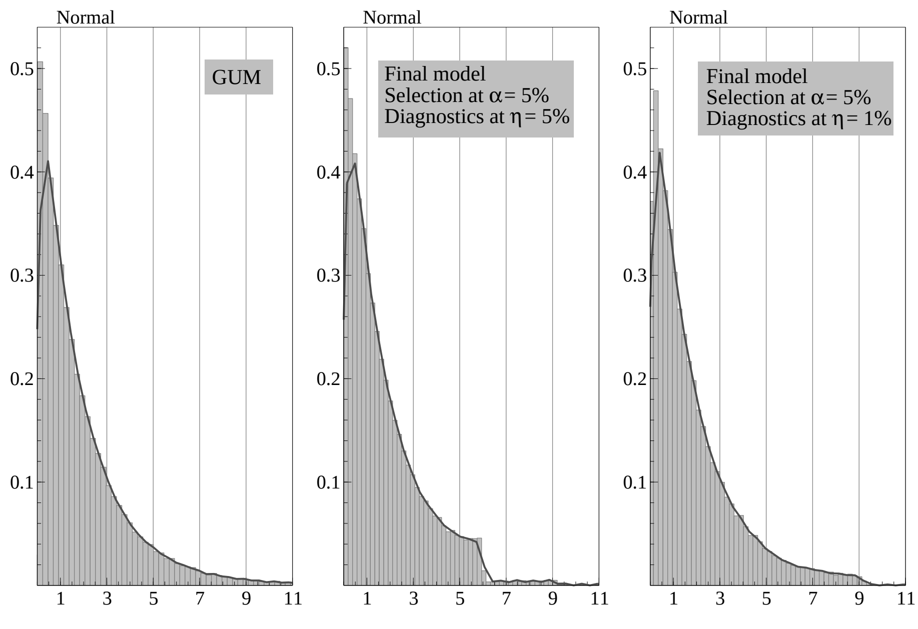

Figure 9
Distribution of normality test, before and after selection. $T = 100, M = 10^5$.

Table 3
Effect of model selection on distribution of diagnostic tests: rejection frequencies in the right tail. $M = 100\,000,\ T = 100$. α is the nominal significance level of *Autometrics*, η nominal significance level of diagnostic tests.

	10%	5%	2.5%	1%	0.5%
GUM					
AR(2)	0.0948	0.0467	0.0229	0.0090	0.0047
ARCH(1)	0.0763	0.0337	0.0158	0.0070	0.0039
Normal	0.0948	0.0506	0.0282	0.0136	0.0081
Hetero	0.0913	0.0500	0.0293	0.0143	0.0080
ChowF	0.0981	0.0497	0.0256	0.0101	0.0050
Selected model $\alpha = 5\%, \eta = 1\%$					
AR(2)	0.0798	0.0350	0.0153	0.0011	0.0008
ARCH(1)	0.0775	0.0351	0.0153	0.0016	0.0012
Normal	0.0919	0.0482	0.0239	0.0042	0.0034
Hetero	0.0940	0.0529	0.0283	0.0015	0.0012
ChowF	0.0972	0.0483	0.0236	0.0034	0.0024
Selected model $\alpha = 1\%, \eta = 1\%$					
AR(2)	0.0897	0.0419	0.0181	0.0009	0.0007
ARCH(1)	0.0770	0.0353	0.0158	0.0011	0.0009
Normal	0.0899	0.0469	0.0232	0.0033	0.0027
Hetero	0.0979	0.0550	0.0285	0.0010	0.0008
ChowF	0.0958	0.0478	0.0232	0.0028	0.0021
Selected model $\alpha = 5\%, \eta = 5\%$					
AR(2)	0.0718	0.0091	0.0059	0.0012	0.0009
ARCH(1)	0.0677	0.0091	0.0057	0.0018	0.0013
Normal	0.0826	0.0176	0.0124	0.0044	0.0036
Hetero	0.0790	0.0072	0.0052	0.0016	0.0012
ChowF	0.0946	0.0220	0.0155	0.0037	0.0025

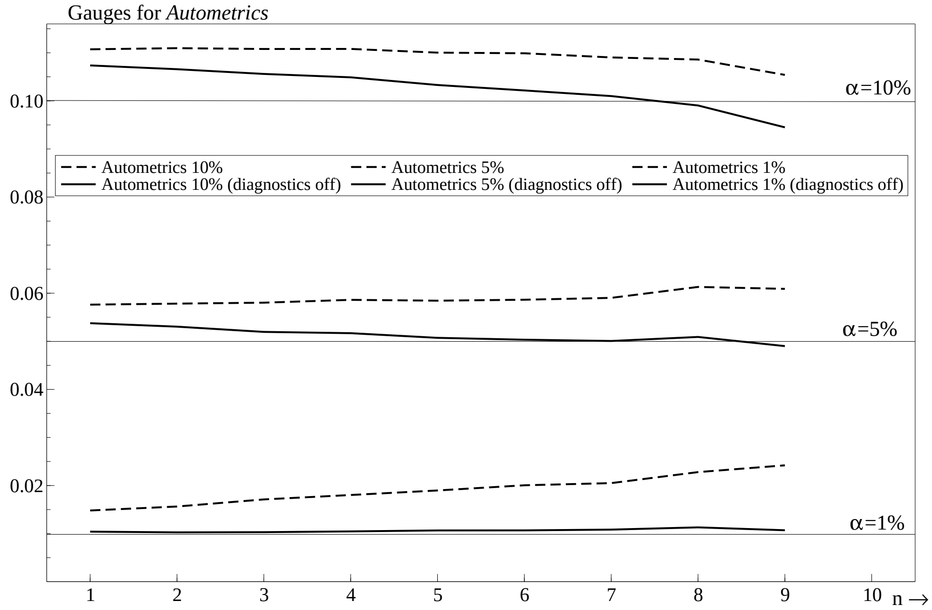

Figure 10
Gauges for *Autometrics* with diagnostic tracking at $\eta = 0.01$ (dashed lines) and without (solid lines) for $\alpha = 0.01, 0.05, 0.1$. The horizontal axis represents the $n = 1, \ldots, 10$ DGPs, each with n relevant variables (and a further $10 - n$ irrelevant in the GUM).

12.3 Simulating *Autometrics* with diagnostic tracking

We return to the Castle et al. experiment, still using $T = 75$ and $M = 10000$ replications. The results were generated in section 11.5. See Figure 10 and Figure 11.

12.4 Impact of diagnostic tracking on MSEs

The results were generated in section 11.5. See Figure 12.

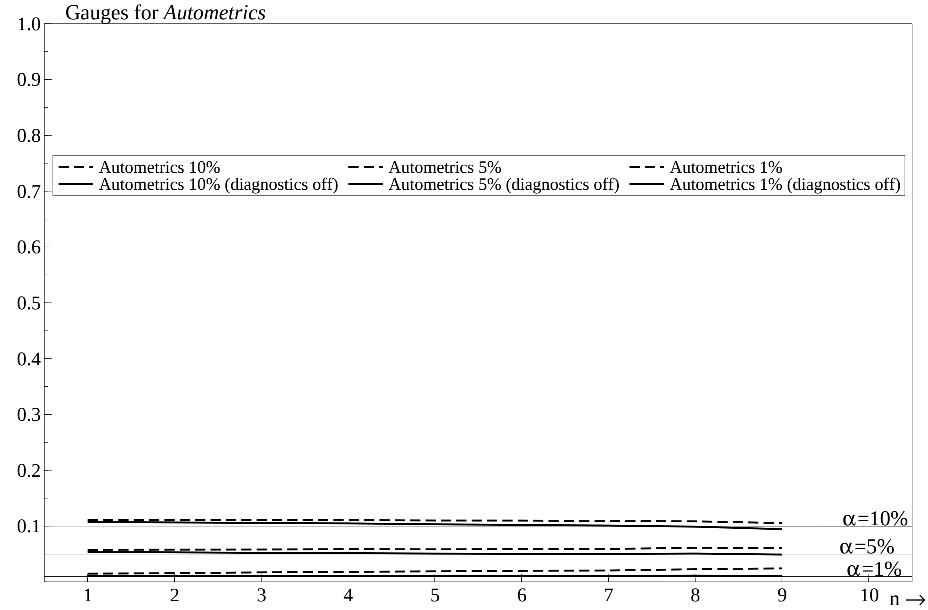

Figure 11
Gauges for *Autometrics* with diagnostic tracking at $\eta = 0.01$ (dashed lines) and without (solid lines) for $\alpha = 0.01, 0.05, 0.1$. The horizontal axis represents the $n = 1, \ldots, 10$ DGPs, each with n relevant variables (and $10 - n$ irrelevant).

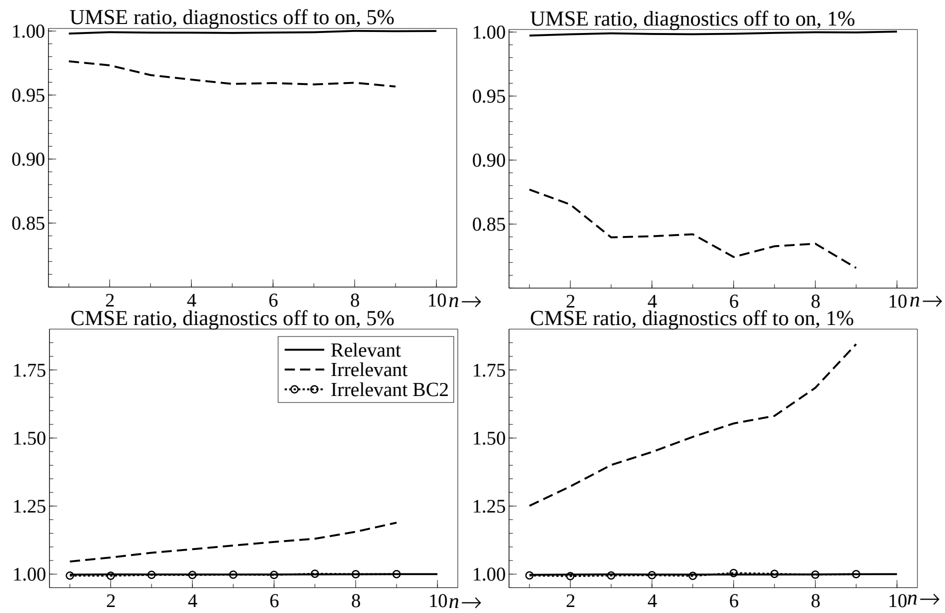

Figure 12
Ratios of MSEs with diagnostic tests off to on for unconditional and conditional distributions

13 Role of Encompassing

13.3 Encompassing the GUM

HP DGP

See chapter 17.

HP80 has $\lambda = 50$, HP800 has $\lambda = 10$.

HP GUM

See chapter 17.

Requirements

Program	`13_03_encompassing.ox`, `13_03_encompassing_step.ox`
Software	Ox 7
Dependencies	`13_03_encompassing.ox`: PcGive and PcGiveExp classes
	`13_03_encompassing_step.ox`: PcGive and AutometricsExp classes
Also uses	`sim_autometrics.ox`, `simdesign.ox`, `simdesigns.ox`, `simstore.ox`
Data	`HooverPerez(1999).xls`
Output	`13_03_encompassing.out`, `13_03_encompassing_step.out`
Running time	about an hour and a half (a few minutes for `13_03_encompassing_step.ox`)

Results

The results for table 13.1 are obtained in chapter 17 below. The results for table 13.2 are obtained from `13_03_encompassing.ox` and `13_03_encompassing_step.ox`:

		1. *Autometrics* encompassing		2. *Autometrics* no encompassing		3. Step-wise regression		4. Backward elimination	
α	λ	gauge	potency	gauge	potency	gauge	potency	gauge	potency
0.1	50	0.093	0.434	0.055	0.394	0.073	0.422	0.192	0.524
0.05	50	0.056	0.406	0.021	0.360	0.039	0.389	0.106	0.455
0.01	50	0.014	0.354	0.002	0.337	0.009	0.348	0.021	0.366
0.1	10	0.097	0.942	0.061	0.902	0.079	0.891	0.197	0.933
0.05	10	0.058	0.937	0.031	0.832	0.046	0.817	0.110	0.923
0.01	10	0.017	0.904	0.018	0.623	0.021	0.696	0.029	0.902
0.1	1	0.093	1.000	0.048	1.000	0.094	0.793	0.184	1.000
0.05	1	0.057	1.000	0.020	1.000	0.061	0.693	0.100	1.000
0.01	1	0.014	1.000	0.002	0.999	0.031	0.533	0.020	1.000

13.4 Iteration and encompassing

The results for table 13.3 are obtained from `13_03_encompassing.ox`:

Autometrics, without pre-search and diagnostics					
		Encompassing GUM_0		Encompassing Intermediate GUM	
α	λ	Gauge	Potency	Gauge	Potency
0.1	50	0.093	0.434	0.138	0.477
0.05	50	0.056	0.406	0.078	0.422
0.01	50	0.014	0.354	0.015	0.357
0.1	10	0.097	0.942	0.135	0.950
0.05	10	0.058	0.937	0.084	0.944
0.01	10	0.017	0.904	0.022	0.909
0.1	1	0.093	1.000	0.130	1.000
0.05	1	0.057	1.000	0.075	1.000
0.01	1	0.014	1.000	0.014	1.000

15 Detecting Outliers and Breaks Using IIS

15.5 Impulse-indicator saturation in *Autometrics*

Castle et al. DGP

See chapter 11.

Castle et al. GUM

See chapter 11.

As before, using $n = 10$, this defines 10 experiments. $T = 75$, $M = 10^5$, z_t fixed, constant forced in GUM.

Requirements

Program	`15_05_1cut_iis.ox` (no IIS, with IIS, $M = 10000$)
Software	Ox 7
Dependencies	PcGive class, `sim_1cut.ox`
Also uses	`simutils.ox`
Output	`15_05_1cut_iis.out` (Autometrics, no diagnostics, with and without IIS)
Running time	about 8.5 hours

Results

See Figure 13 and table 15.1:

α	1%		0.1%	
	no IIS	IIS	no IIS	IIS
average gauge (variables)	1.06%	1.46%	0.10%	0.10%
average gauge (variables and dummies)	—	1.48%	—	0.09%
average potency	99.99%	99.98%	99.90%	99.86%

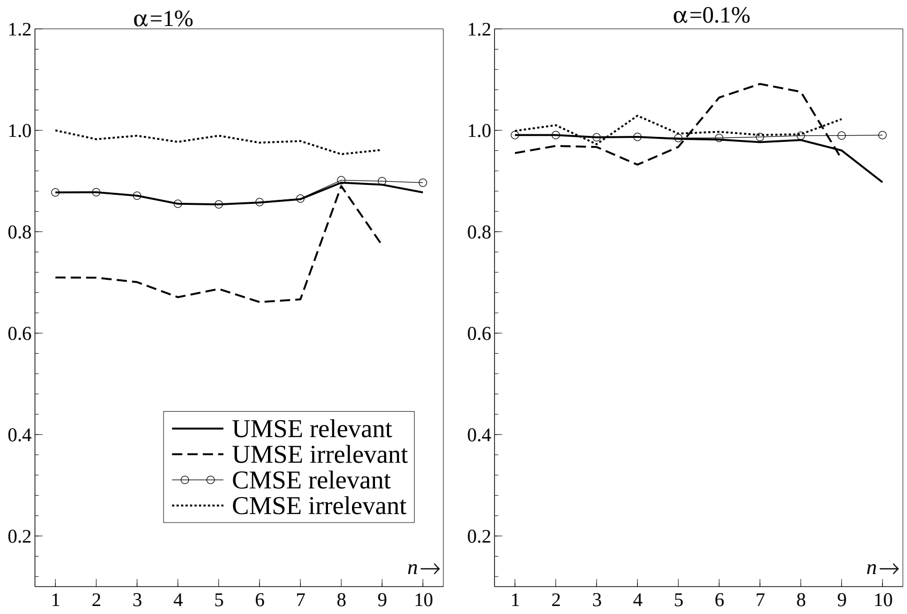

Figure 13
Ratios of MSEs without IIS to with.

15.6 IIS in a fat-tailed distribution

Castle et al. DGP

Adjusted from chapter 11 to:

$$
\begin{aligned}
y_t &= 5 + \sum_{k=1}^{n} z_{k,t} + \epsilon_t, & \epsilon_t &\sim 0.4 n^{1/2} t_3 \\
z_t &= (z_{1,t}, \ldots, z_{10,t})', & z_t &\sim \mathsf{IN}_{10}\left[\mathbf{0}, \boldsymbol{I}_{10}\right].
\end{aligned}
$$

Castle et al. GUM

See chapter 11.

As before, using $n = 10$, this defines 10 experiments. $T = 75$, $M = 10^5$, z_t fixed, constant forced in GUM.

Requirements

Program	`15_06_1cut_iis_t3.ox` (without and with IIS, with and without diagnostics)
	`15_06_1cut_iis_t3_plot.ox` (plots the case $n = 5, \alpha = 0.1$))
Software	Ox 7
Dependencies	PcGive class, `sim_1cut.ox`
Also uses	`simutils.ox`
Output	`15_06_iis_t3.out`
Running time	about 11 hours (45 minutes for `15_06_1cut_iis_t3_plot.ox`)

Results

See Figure 14 and Figure 15 below, and table 15.2:

IIS diagnostic tracking	no yes	no no	yes no	no yes	no no	yes no
α		1%			0.1%	
average gauge% (variables)	8.05	1.19	1.52	6.39	0.31	0.14
average gauge% (variables and dummies)	–	–	5.25	–	–	0.17
average potency%	96.26	95.58	98.78	92.42	91.01	90.80

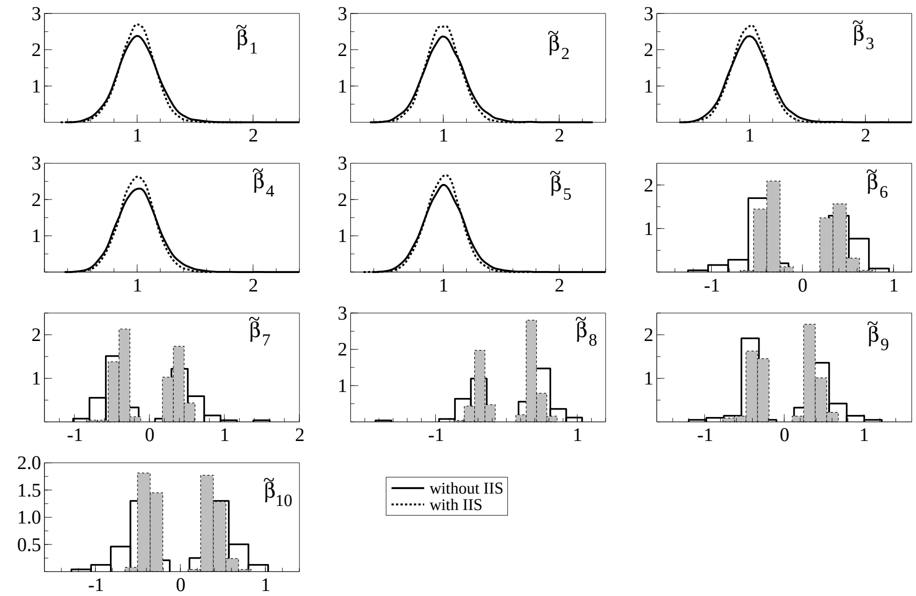

Figure 14
Conditional distributions of estimates with (dark) and without (light) IIS for the $n = 5$ experiment with a t_3 error distribution at $\alpha = 1\%$, $M = 10\,000$.

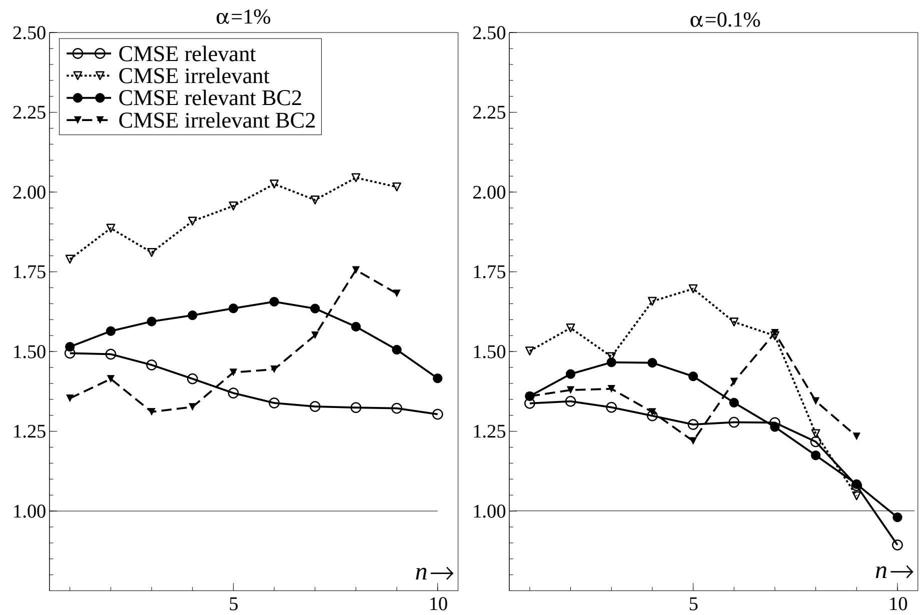

Figure 15
Ratios of MSEs without IIS to IIS for a t_3-distribution with no diagnostic testing.

15.9 Impulse-indicator saturation simulations

D_i DGP

D_1:	$y_{1,t} = \lambda \left(I_{T-19} + \cdots + I_T\right) + u_t,$	$u_t \sim \mathsf{N}[0,1],$
D_2:	$y_{2,t} = \lambda \left(I_1 + \ldots + I_{20}\right) + u_t,$	$u_t \sim \mathsf{N}[0,1],$
D_3:	$y_{3,t} = \lambda \left(I_1 + I_6 + I_{11} + \cdots\right) + u_t,$	$u_t \sim \mathsf{N}[0,1].$

$T = 100.$

D_i GUM

$$y_t = \gamma_0^F + \sum_{k=1}^{T} \gamma_k I_k + u_t.$$

Requirements

Program	`15_09_breaks.ox`
Software	Ox 7
Dependencies	PcGive and PcGiveExp classes
Also uses	`sim_pcgive.ox`, `simdesign.ox`, `simdesigns.ox`, `simstore.ox`
Output	`15_09_breaks.out`
Running time	about 25 minutes

Results

	Autometrics at 1%			
	D_1		D_3	
	$\lambda = 3$	$\lambda = 4$	$\lambda = 3$	$\lambda = 4$
Gauge %	0.4	0.7	0.3	1.1
Potency %	54.8	86.3	24.1	65.9

16 Re-modeling UK Real Consumers' Expenditure

16.2 Replicating DHSY

Requirements

Program	`16_02_DHSY_45aa.fl`
Data	`DHSY.xlsx`
Software	PcGive
Output	`16_02_DHSY_45aa.out`

Results

The OxMetrics batch file replicates the simultaneous equations model that corresponds to the DGP.

DHSY

The `DHSY.xlsx` data set contains the data that were used by DHSY, Davidson, Hendry, Srba, and Yeo (1978). The following Algebra code:

```
c = LC;
y = LY;
D4p = D4LPC;
D4c = diff(c, 4);
D4y = diff(y, 4);
DD4p = diff(D4p, 1);
DD4y = diff(D4y, 1);
"c-y" = LC - LY;
D = D6812 + D7312;
D4D = diff(D, 4);
```

was used in OxMetrics to create the variables:

c	$c_t,$	D4c	$\Delta_4 c_t,$
y	$y_t,$	DD4y	$\Delta\Delta_4 y_t,$
D4p	$\Delta_4 p_t,$	DD4p	$\Delta\Delta_4 p_t,$
c-y	$(c-y)_t,$		
D	$D_t,$		
D4D	$\Delta_4 D_t,$		

First, `16_02_DHSY_45aa.fl` replicates (45)** exactly:

$$\Delta_4 c_t = \underset{(0.03)}{0.48}\Delta_4 y_t - \underset{(0.04)}{0.23}\Delta_1\Delta_4 y_t - \underset{(0.02)}{0.12}\Delta_4 p_t - \underset{(0.01)}{0.31}\Delta_1\Delta_4 p_t + \underset{(0.002)}{0.006}\Delta_4 D_t - \underset{(0.01)}{0.09}\left(c - y\right)_{t-4}$$

$\widehat{\sigma} = 0.0062 \quad T = 67 : 1959(2) - 1975(4).$

Equation (16.1) in the book uses one additional observation at the start:

$$\Delta_4 c_t = \underset{(0.029)}{0.479}\Delta_4 y_t - \underset{(0.039)}{0.230}\Delta_1\Delta_4 y_t - \underset{(0.022)}{0.119}\Delta_4 p_t - \underset{(0.010)}{0.302}\Delta_1\Delta_4 p_t + \underset{(0.0022)}{0.00645}\Delta_4 D_t - \underset{(0.012)}{0.0940}\left(c - y\right)_{t-4}$$

$\widehat{\sigma} = 0.00613 \quad T = 68 : 1959(1) - 1975(4).$

16.3 Selection based on *Autometrics*

Requirements

Program	`16_03_DHSY_IIS.fl`, `16_03_DHSY_IIS_ECM.fl`
Data	`DHSY.xlsx`
Software	PcGive
Output	`16_03_DHSY_IIS.out`, `16_03_DHSY_IIS_ECM.out`

Results

`16_03_DHSY_IIS.fl` first estimates (16.3), with the GUM (without IIS) formulated as:

```
system
{
    Y = c, c_1, c_2, c_3, c_4, c_5;
    Z = y, y_1, y_2, y_3, y_4, y_5, D4p, D4p_1, D4p_2;
    U = Constant, CSeasonal, CSeasonal_1, CSeasonal_2, D4D;
}
```

IIS finds no further dummies. Next, `16_03_DHSY_IIS.fl` re-estimates without D4D, which again does not find any outliers (and also drops $\Delta_4 p_{t-1}$).

`16_03_DHSY_IIS_ECM.fl` estimates (16.5), finding two additional dummy variables.

17 Comparisons of *Autometrics* with Other Approaches

17.2.1 HP: US macroeconomic data experiments

HP DGP

DGP	design
HP2	$y_t = 0.75y_{t-1} + 85.99\varepsilon_t$
HP7	$y_t = 0.75y_{t-1} + 1.33x_{11,t} - 0.9975x_{11,t-1} + 6.44\varepsilon_t$
HP8	$y_t = 0.75y_{t-1} - 0.046x_{3,t} + 0.0345x_{3,t-1} + 0.073\varepsilon_t$
HP8(λ)	$y_t = 0.75y_{t-1} - 0.046x_{3,t} + 0.0345x_{3,t-1} + 0.073\lambda\varepsilon_t$
HP9	$y_t = 0.75y_{t-1} - 0.023x_{3,t} + 0.01725x_{3,t-1} + 0.67x_{11,t} - 0.5025x_{11,t-1} + 3.25\varepsilon_t$

HP GUM

$$y_t = \gamma_0^F + \sum_{j=1}^{4} \alpha_j y_{t-j} + \sum_{i=1}^{18} \sum_{j=0}^{1} \gamma_{i,j} x_{i,t-j} + u_t \text{ where } u_t \sim \mathsf{IN}\left[0, \sigma_u^2\right]. \quad (1)$$

Requirements

Program	`17_02_pcnaive_HP#.ox`
Software	Ox 7
Dependencies	PcNaive and PcGive class
Data	`HooverPerez(1999).xls`
Output	`17_02_pcnaive_HP#.out`
Running time	less than a minute

The code for these experiments was first created with PcNaive. HP2 does not have regressors, but the other experiments do. So they have been adjusted to load the data set, and then use that as follows:

```
CMyPcFimlDgp::GenerateZ(const cT, const mC0t, const mV)
{
    decl db = new Database();
    db.Load("../data/HooverPerez(1999).xls");
    db.Info();
    // x3 and x11
    decl mz = diff0(diff0(db.GetVar("GGEQ"))) ~ diff0(db.GetVar("FM1DQ"));
    delete db;
    return mz;
    // 5 discarded will line up data to use 1960(3) - 1995(1) in estimation
}
```

The code for HP8 has a lambda variable in `CPcNaiveExp::CPcNaiveExp()` to simulate HP8(λ) in section 13.3. The value of lambda is noted in the output.

Results

These experiments provide the means of the t-tests and R^2s reported in Table 4.

Table 4
Small sample properties of some HP experiments, $T = 139$ and $M = 100\,000$

DGP	T	n	R^2	t-values
HP2	139	1	0.53	12.7
HP7	139	3	0.81	12.4, 15.2, −8.2
HP8	139	3	0.97	12.7, −58.8, 12.1
HP9	139	5	0.81	12.4, −0.7, 0.5, 15.2, −8.2

17.3 Re-analyzing the Hoover–Perez experiments

Requirements

Program	`17_03_hp.ox`
Software	Ox 7
Dependencies	PcGive and PcGiveExp classes
Also uses	`sim_pcgive.ox`, `simdesign.ox`, `simdesigns.ox`, `simstore.ox`
Data	`HooverPerez(1999).xls`
Output	`17_03_hp.out`
Running time	more than an hour

Presearch is switched off with:

```
exp.Autometrics(pvalue[j], "none", 0);
```

Using default settings:

```
exp.Autometrics(pvalue[j]);
```

Default settings, except that effort is set to zero (labelled tight in the table):

```
            exp.Autometrics(pvalue[j], "none", 1);
            exp.AutometricsSet("effort", 0);
```

Results

The PcGets and Hoover-Perez results reported in table 17.3 are taken from Doornik (2009, table 6). That table reports Autometrics results with chopping and bunching switched off. The new results use default settings, except that presearch is switched off:

	Hoover–Perez				*PcGets*				*Autometrics*			
	HP2	HP7	HP8	HP9	HP2	HP7	HP8	HP9	HP2	HP7	HP8	HP9
1% nominal significance level												
gauge	5.7	3.0	0.9	3.2	2.4	2.4	–	2.5	1.5	1.5	1.5	1.5
potency	100	94.0	99.9	57.3	100	99.9	–	61.9	100	99.7	100	60.6
DGPf	0.8	24.6	78.0	0.8	60.2	59.0	–	0.0	68.6	69.6	69.0	0.0
5% nominal significance level												
gauge	10.7	8.2	3.7	8.5	10.7	10.2	–	10.4	5.6	5.7	5.8	5.8
potency	100	96.7	100	60.4	100	99.9	–	66.2	100	99.9	100	62.8
DGPf	0.0	4.0	31.6	1.2	8.4	4.0	–	0.0	16.3	18.4	17.9	0.3
10% nominal significance level												
gauge	16.2	14.2	10.6	14.1	–	–	–	–	9.2	9.6	9.3	9.5
potency	100	96.9	100	62.5	–	–	–	–	100	99.9	100	64.1
DGPf	0.0	0.2	7.6	0.4	–	–	–	–	3.3	2.7	3.1	0.2

	PcGets		*Autometrics*						
	Default		Default				Tight		
	HP2	HP7	HP2	HP7	HP8	HP9	HP2	HP7	HP8
1% nominal significance level									
gauge %	0.9	1.0	1.2	1.6	1.6	1.6	0.9	1.0	0.9
potency %	100	99.8	100	99.0	100	60.2	100	99.1	100
DGPf	81.0	80.8	73.5	68.0	68.5	0.0	79.0	80.0	79.5
5% nominal significance level									
gauge %	5.5	5.4	5.1	5.9	5.9	6.1	3.8	4.1	4.0
potency %	100	99.8	100	99.7	100	62.6	100	99.6	100
DGPf	34.5	34.7	28.2	17.0	17.9	0.3	39.4	38.9	38.7

17.4 Step-wise regression comparisons

Requirements

Program	`17_04_cmp.ox`
Software	Ox 7
Dependencies	PcFimlEx and AutometricsExp classes
Also uses	`sim_autometrics.ox`, `simdesign.ox`, `simdesigns.ox`, `simstore.ox`
Data	`HooverPerez(1999).xls`
Output	`17_04_cmp.out`
Running time	about half an hour

The PcGive class contains Autometrics, but does not have the ability to select a model by step-wise regression, lasso or backward selection. Therefore these experiments use the Autometrics class, together with PcFimlEx for estimation.

To run step-wise regression:

```
            exp.Stepwise("Stepwise", pvalue[j]);
```

Results

In addition to gauge and potency, table 17.5 reports the percentage of replications in which the final model exactly equals the DGP (labelled DGP found), as well as the percentage in which the final model contains the DGP (DGP nested).

	HP7	HP8	HP7	HP8
	Step-wise		*Autometrics*	
***1%* nominal significance**				
gauge %	0.9	3.1	1.6	1.6
potency %	100	53.3	99.0	100
DGP found %	71.6	22.0	68.0	68.5
DGP nested %	100	30.0	97.1	100
***5%* nominal significance**				
gauge %	3.8	6.1	5.9	5.9
potency %	100	69.3	99.7	100
DGP found %	26.8	13.5	17.0	17.9
DGP nested %	99.9	53.9	99.2	100
0.1% nominal significance				
gauge %	0.1	1.9	0.9	0.3
potency %	99.9	40.5	97.1	100
DGP found %	95.1	10.5	85.0	92.9
DGP nested %	99.8	10.8	91.4	100

17.6 Lasso

JEDC(ρ) DGP

$$\begin{array}{lcl} y_t & = & \sum_{i=1}^{5} \beta_i x_{i,t} + \epsilon_t, \quad \text{where} \quad \epsilon_t \sim \mathsf{IN}\left[0,1\right] \\ \boldsymbol{x}_t & \sim & \mathsf{IN}_{10}\left[0, \boldsymbol{C}_x\right] \end{array} \tag{2}$$

where $\boldsymbol{x}_t' = (x_{1,t}, ..., x_{10,t})$ and the elements $c_{i,j}$ of the correlation matrix $\boldsymbol{C}_x$ are specified as $c_{i,i} = 1$ and:

$c_{i,j} = \rho^{|i-j|}$.

Finally, we specify the coefficients as:

$\beta_1 = 8/\sqrt{T}, \beta_2 = 6/\sqrt{T}, \beta_3 = 4/\sqrt{T}, \beta_4 = 3/\sqrt{T}, \beta_5 = 2/\sqrt{T}$.

JEDC DGP

The JEDC DGP equals JEDC($\rho = 0$)

JEDC-EJ(ρ) DGP

$$\begin{array}{lcl} y_t & = & \sum_{i=1}^{5} \beta_i x_{i,t} + \epsilon_t, \quad \text{where} \quad \epsilon_t \sim \mathsf{IN}\left[0,1\right] \\ \boldsymbol{x}_t & = & \rho \boldsymbol{x}_{t-1} + v_t, \quad \text{where} \quad \mathsf{IN}_{10}\left[0, (1-\rho^2)\boldsymbol{I}_{10}\right]. \end{array} \tag{3}$$

JEDC and JEDC-EJ GUM

The GUM has a forced intercept and $N = 21$ free variables:

$$y_t = \gamma_0^F + y_{t-1} + \sum_{i=1}^{10} \sum_{j=0}^{1} \gamma_{i,j} x_{i,t-j} + u_t \text{ where } u_t \sim \mathsf{IN}\left[0, \sigma_u^2\right]. \tag{4}$$

So there are $n = 5$ relevant and $m = 16$ irrelevant regressors.

Requirements

Program	HP: `17_06_cmp1.ox`
	JEDC-EJ: `17_06_cmp2.ox`
	JEDC: `17_06_cmp3.ox`
Software	Ox 7
Dependencies	PcFimlEx and AutometricsExp classes
Also uses	`sim_autometrics.ox`, `simdesign.ox`, `simdesigns.ox`, `simstore.ox`
Data	`HooverPerez(1999).xls` (`17_06_cmp1.ox` only)
Output	`17_06_cmp1.out`, `17_06_cmp2.out`, `17_06_cmp3.out`
Running time	about half an hour for `17_06_cmp1.ox`, a few minutes for the others

To run Lasso until the end, selecting the best model using BIC:

```
        exp.Stepwise("Lasso", 0, -1, 3);        // BIC
```

To run Lasso selecting a model by size:

```
        exp.Stepwise("Lasso", 0, truesize[i], -1);
```

To run Autometrics selecting a model by size:

```
        exp.Stepwise("Auto_size", 0, truesize[i], -1);
```

Results

The results are in tables 17.6 and 17.7 and Figure 16.

	Lasso selection BIC				*Autometrics* 5% nominal significance			
	HP2	HP7	HP8	HP9	HP2	HP7	HP8	HP9
gauge %	2.2	19.5	35.1	18.6	5.1	5.9	5.9	6.1
potency %	100	94.4	86.3	65.6	100	99.7	100	62.6
DGP found %	53.1	0.1	0.0	0.0	28.2	17.0	17.9	0.3
DGP nested %	100	83.2	68.1	7.4	100	99.2	100	1.3
	Lasso selection true n				*Autometrics* 1% nominal significance			
	HP2	HP7	HP8	HP9	HP2	HP7	HP8	HP9
gauge %	0.0	2.7	3.3	7.9	1.2	1.6	1.6	1.6
potency %	100	66.8	59.3	44.5	100	99.0	100	60.2
DGP found %	100	0.5	0.0	0.0	73.5	68.0	68.5	0.0
DGP nested %	100	0.5	0.0	0.0	100	97.1	100	0.4
	Lasso selection true n				*Autometrics* true n			
	HP2	HP7	HP8	HP9	HP2	HP7	HP8	HP9
gauge %	0.0	2.7	3.3	7.9	0.0	0.1	0.0	5.4
potency %	100	66.8	59.3	44.5	100	98.8	100	62.5
DGP found %	100	0.5	0.0	0.0	100	96.4	100	0.8
DGP nested %	100	0.5	0.0	0.0	100	96.4	100	0.8

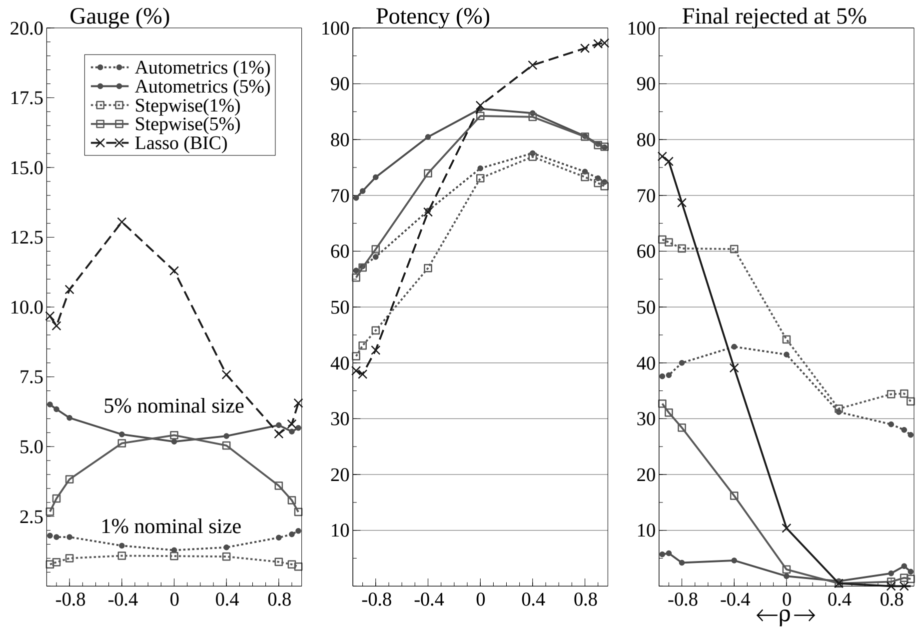

Figure 16
Correlation range experiments JEDC(ρ)

	Step-wise		Lasso		*Autometrics*			
	BIC	BIC	BIC	BIC	5%	5%	11%	15%
ρ	*0.2*	*0.8*	*0.2*	*0.8*	*0.2*	*0.8*	*0.2*	*0.8*
gauge %	4.5	5.7	11.3	16.5	5.3	6.3	11.1	15.2
potency %	82.5	70.6	85.8	76.9	85.2	76.5	89.5	81.0
DGP found%	16.4	7.0	7.3	0.9	20.9	13.1	14.5	7.2
DGP nested %	31.5	11.5	46.1	27.7	38.1	21.4	53.4	30.8
Final rej. at 5%	8.6	7.4	10.7	13.2	1.9	3.6	0.2	0.1
	Step-wise		Lasso		*Autometrics*			
	true n	true n	true n	true n	5%	10%		
ρ	*0.2*	*0.8*	*0.2*	*0.8*	*0.2*	*0.8*		
gauge %	4.8	8.1	5.5	10.0	5.3	10.9		
potency %	84.6	74.0	82.5	67.8	85.2	79.2		
DGP found%	33.9	12.3	26.3	3.8	20.9	9.7		
DGP nested%	33.9	12.3	26.3	3.8	38.1	26.9		
Final rej. at 5%	10.0	9.0	21.4	40.6	1.9	0.5		

17.7 Comparisons with RETINA

DGP for Nonlinear GUM

$$
\begin{array}{rcll}
y_t & = & \sum_{i=1}^{J} \beta_i x_{i,t} + \epsilon_t, & \epsilon_t \sim \mathsf{IN}[0,1] \\
\boldsymbol{x}_t & = & 10 + \boldsymbol{\nu}_t, & \boldsymbol{\nu}_t \sim \mathsf{IN}_2[0, \boldsymbol{I}_2]
\end{array}
$$

for $t = 1, \ldots, T$ where $\boldsymbol{x}_t = (x_{1,t}, x_{2,t})'$ and $T = 100$. Two DGPs are defined, with either one or two variables: $J = 1$ or $J = 2$. The DGP is from Castle (2005). Specifically, we use:

$$
\begin{array}{llcll}
(a) & y_t & = & 0.4x_{1,t} + \epsilon_t, & \epsilon_t \sim \mathsf{IN}[0,1], \\
(b) & y_t & = & 1100x_{1,t} + \epsilon_t, & \epsilon_t \sim \mathsf{IN}[0,1], \\
(c) & y_t & = & 800x_{1,t} + 1600x_{2,t} + \epsilon_t, & \epsilon_t \sim \mathsf{IN}[0,1].
\end{array}
$$

Nonlinear GUM

Substantial collinearity is generated between regressors by using the following transformations in the GUM:

$$
y_t = \gamma + \gamma_1 x_{1,t} + \gamma_2 x_{2,t} + \gamma_3 x_{1,t}^2 + \gamma_4 x_{2,t}^2 + \gamma_5 x_{1,t}^{-1} + \gamma_6 x_{2,t}^{-1} + \gamma_7 x_{1,t}^{-2} + \gamma_8 x_{2,t}^{-2} + \gamma_9 x_{1,t} x_{2,t} + \gamma_{10} x_{1,t}^{-1} x_{2,t}^{-1}
$$
$$
+ \gamma_{11} x_{1,t} x_{2,t}^{-1} + \gamma_{12} x_{2,t} x_{1,t}^{-1} + \varepsilon_t
$$

DGP (b), $J = 1$ with $\beta_1 = 1100$, corresponds to $\mathsf{t}_{\beta_1} \approx 4$ in the GUM. DGP (c), $J = 2$ with $\beta_1 = 800$ and $\beta_2 = 1600$, corresponds to $\mathsf{t}_{\beta_1} \approx 3$, and $\mathsf{t}_{\beta_2} \approx 6$ in the GUM.

We also consider the gum after double demeaning (DDM). In that case, every transformation involves a zero mean variable, and is demeaned itself as well:

$$
y_t = \gamma + \gamma_1 x_{1,t} + \gamma_2 x_{2,t} + \gamma_3 \overline{z_{1,t}^2} + \gamma_4 \overline{z_{2,t}^2} + \gamma_5 \overline{z_{1,t}^{-1}} + \gamma_6 \overline{z_{2,t}^{-1}} + \gamma_7 \overline{z_{1,t}^{-2}} + \gamma_8 \overline{z_{2,t}^{-2}} + \gamma_9 \overline{z_{1,t} z_{2,t}} + \gamma_{10} \overline{z_{1,t}^{-1} z_{2,t}^{-1}}
$$
$$
+ \gamma_{11} \overline{z_{1,t} z_{2,t}^{-1}} + \gamma_{12} \overline{z_{2,t} z_{1,t}^{-1}} + \varepsilon_t,
$$

where $z_{i,t} = \overline{x_{i,t}} = x_{i,t} - T^{-1} \sum_{t=1}^{T} x_{i,t}$.

Requirements

Program $J = 1, \beta = 0.4$	`17_07_dgpa_sel1.ox` (1%), `17_07_dgpa_sel5.ox` (5%)
Program $J = 1$	`17_07_dgpb_gum.ox` (GUM only), `17_07_dgpb_sel1.ox` (1%), `17_07_dgpb_sel5.ox` (5%), `17_07_dgpb_sel1_ddm.ox` (1%, double demeaned), `17_07_dgpb_sel5_ddm.ox` (5%, double demeaned)
Program $J = 2$	`17_07_dgpc_gum.ox` (GUM), `17_07_dgpc_sel1.ox` (1%), `17_07_dgpc_sel5.ox` (5%)
Software	Ox 7
Dependencies	PcGive and PcNaive classes
Output	`17_07_dgp*.out`
Running time	all less than a minute

The experiments are implemented in PcNaive, using custom Z's:

```
CPcNaiveExp::TransformZ(const mZ)
{
    Renew( sqr(mZ[][0]), "CZ0");
    Renew( sqr(mZ[][1]), "CZ1");
    Renew( 1 ./ mZ[][0], "CZ2");
    Renew( 1 ./ mZ[][1], "CZ3");
    Renew( 1 ./ sqr(mZ[][0]), "CZ4");
    Renew( 1 ./ sqr(mZ[][1]), "CZ5");
    Renew( mZ[][0] .* mZ[][1], "CZ6");
    Renew( 1 ./ (mZ[][0] .* mZ[][1]), "CZ7");
    Renew( mZ[][0] ./ mZ[][1], "CZ8");
    Renew( mZ[][1] ./ mZ[][0], "CZ9");
}
```

Double demeaning (DDM) is implemented as:

```
CPcNaiveExp::TransformZ(const mZ)
{
    decl z0 = mZ[][0], z1 = mZ[][1], zcz;
    z0 -= meanc(z0);
    z1 -= meanc(z1);
    zcz = sqr(z0);          Renew(zcz - meanc(zcz) , "CZ0");
    zcz = sqr(z1);          Renew(zcz - meanc(zcz) , "CZ1");
    zcz = 1 ./ z0;          Renew(zcz - meanc(zcz) , "CZ2");
    zcz = 1 ./ z1;          Renew(zcz - meanc(zcz) , "CZ3");
    zcz = 1 ./ sqr(z0);     Renew(zcz - meanc(zcz) , "CZ4");
    zcz = 1 ./ sqr(z1);     Renew(zcz - meanc(zcz) , "CZ5");
    zcz = z0 .* z1;         Renew(zcz - meanc(zcz) , "CZ6");
    zcz = 1 ./ (z0 .* z1);  Renew(zcz - meanc(zcz) , "CZ7");
    zcz = z0 ./ z1;         Renew(zcz - meanc(zcz) , "CZ8");
    zcz = z1 ./ z0;         Renew(zcz - meanc(zcz) , "CZ9");
}
```

Results

The PcGets and RETINA results are taken from Castle (2005, table 9).

	RETINA	PcGets		Autometrics	
		Lib.	Cons.	5%	1 %
Gauge (%)					
$J = 1, \beta_1 = 0.4$	10.0	25.4	14.2	8.9	7.1
$J = 1, \beta_1 = 1100$	0.3	15.3	4.4	4.6	1.1
$J = 2, \beta_1 = 800, \beta_2 = 1600$	66.6	16.5	6.8	4.7	1.4
$J = 1, \beta_1 = 1100$, double demeaned	2.6	5.2	1.2	5.3	1.3
Potency (%)					
$J = 1, \beta_1 = 0.4$	6.6	33.1	19.0	61.1	58.7
$J = 1, \beta_1 = 1100$	100	98.0	98.4	99.6	99.6
$J = 2, \beta_1 = 800, \beta_2 = 1600$	98.7	97.0	97.0	99.6	99.7
$J = 1, \beta_1 = 1100$, double demeaned	81.2	97.0	87.0	100	100

Autometrics performs substantially better than PcGets and RETINA: it is at least as good or, in case of $J = 1, \beta_1 = 0.4$, much better, in picking up relevant variables. At the same time, fewer irrelevant variables are retained, reflected in a gauge that is much closer to the nominal significance level. Doornik (2009, table 8) gives a similar example where Autometrics outperforms PcGets.

Note that the intercept is free (can be removed) in PcGets and Autometrics; forcing the intercept makes the results much worse. RETINA always forces an intercept in the model.

As the table shows, double demeaning improves the gauge of PcGets and RETINA, but at a loss in potency. Autometrics, on the other hand, has improved gauge as well as potency from double demeaning.

18 Model Selection in Underspecified Settings

18.5 A dynamic artificial-data example

Requirements

Program	`18_05_dataz.fl`
Data	`dataz.in7/dataz.bn7`
Software	PcGive
Output	`18_05_dataz.out`

Results

The OxMetrics batch file replicates the estimates in levels, (18.10), in differences, (18.11), and in levels using IIS, (18.12). All are estimated over the same sample period 1955(3) – 1992(3).

19 More Variables than Observations

19.3 Simulation evaluation of alternative block modes

D_i DGP and GUM

See section 15.9.

Requirements

Program	`19_03_breaks.ox`
Software	Ox 7
Dependencies	PcGive and PcGiveExp classes
Also uses	`sim_pcgive.ox`, `simdesign.ox`, `simdesigns.ox`, `simstore.ox`
Output	`19_03_breaks.out`
Running time	about three hours

19.4 Hoover–Perez experiments with $N > T$

HPbig DGP

This is the HP DGP, augmented with 10 IID regressors:

$$\boldsymbol{z}_t \sim \mathsf{IN}_{10}\left[0, \boldsymbol{I}_{10}\right]$$

where $\boldsymbol{z}_t' = (z_{1,t}, \ldots, z_{10,t})$.

HPbig GUM

$$y_t = \gamma_0^F + \sum_{j=1}^{4} \alpha_j y_{t-j} + \sum_{i=1}^{18} \sum_{j=0}^{4} \gamma_{i,j} x_{i,t-j} + \sum_{i=1}^{10} \sum_{j=0}^{4} \delta_{i,j} z_{i,t-j} + u_t \text{ where } u_t \sim \mathsf{IN}\left[0, \sigma_u^2\right]. \tag{5}$$

The dimensions are:

	forced	relevant	irrelevant	T
HP7big	1	3	141	139
HP8big	1	3	141	139

Requirements

Program	`19_04_cmp.ox` (stepwise and Autometrics) `19_04_cmp_pcgive.ox` (Autometrics only)
Software	Ox 7
Dependencies	`19_04_cmp.ox`: PcFimlEx and AutometricsExp classes
Dependencies	`19_04_cmp_pcgive.ox`: PcGive and PcGiveExp classes
Also uses	`sim_autometrics.ox`, `sim_pcgive.ox`, `simdesign.ox`, `simdesigns.ox`, `simstore.ox`
Data	`HooverPerez(1999).xls`
Output	`19_04_cmp.out`, `19_04_cmp_pcgive.out`
Running time	about an hour

Results

Table 19.2 is obtained from `19_04_cmp.ox`:

	Step-wise		*Autometrics*	
	HP7big	HP8big	HP7big	HP8big
1% nominal significance				
gauge %	0.8	1.7	1.3	1.2
potency %	99.9	50.9	96.8	100
DGP found %	32.2	10.0	43.3	47.2
DGP nested %	99.8	26.3	90.5	100
0.1% nominal significance				
gauge %	0.1	0.7	0.3	0.1
potency %	99.7	40.3	97.0	100
DGP found %	87.4	9.0	82.6	90.4
DGP nested %	99.6	10.5	91.2	100

Both `19_04_cmp.ox` and `19_04_cmp_pcgive.ox` run the same HPbig experiments using Autometrics. However, there is a small difference between them: the PcFiml class excludes fixed (U) variables from the heteroscedasticity test, while PcGive includes them. This doesn't matter for variables that are redundant when squared, but does for other variables. So occasionally, a different test p-value will be obtained, and a different model selected. This is most likely to happen in the block search for omitted variables that is used when there are more variables than observations: in that case the current model is kept fixed. There is no difference in the precision that is used in Table 19.2.

19.6 Modeling $N > T$ in practice

Requirements

Program	`19_06_dataz.fl`
Data	`dataz.in7/dataz.bn7`
Software	PcGive
Output	`19_06_dataz.out`

Results

The OxMetrics batch file replicates the estimates in levels from a GUM with lags up to 20 of the four DGP variables, as well as 20 $\mathsf{IIN}[0, 1]$ variables.

20 Impulse-indicator Saturation for Multiple Breaks

20.2 IIS for breaks in the mean of a location-scale model

Break DGPs

DGP:Bc	$y_t = \delta + \gamma\,(I_{81} + \cdots + I_{100}) + u_t,$
DGP:B20	$y_t = \delta + \gamma\,(I_1 + I_6 + I_{11} + \cdots + I_{96}) + u_t,$
DGP:MBc	$y_t = \delta + \gamma\,(I_1 + I_2 + I_3 + I_4 + I_{24} + \cdots + I_{27} + I_{49} + \cdots$
	$+ I_{52} + I_{74} + \cdots + I_{77} + I_{97} + \cdots + I_{100}) + u_t,$
DGP:Bct	$y_t = \delta + \gamma\,(I_{81} + \cdots + I_{100}) + 0.02t + u_t,$
DGP:MBct	$y_t = \delta + \gamma\,(I_1 + I_2 + I_3 + I_4 + I_{24} + \cdots + I_{27} + I_{49} + \cdots$
	$+ I_{52} + I_{74} + \cdots + I_{77} + I_{97} + \cdots + I_{100}) + 0.02t + u_t,$
DGP:Tc	$y_t = \delta + \gamma\left(\frac{1}{20}I_{81} + \frac{2}{20}I_{82} + \cdots + \frac{20}{20}I_{100}\right) + u_t.$
	$u_t \sim \mathsf{IN}[0,1]$; $\delta = 0, 1$ as noted below; $t = 1, \ldots, 100$.

Note that DGP:BLc($\delta = 0$) is the same as DGP:BL.

Break GUMs

Specification of GUM		Used for DGP
GUM:Ic	y_t on 1 & T dummies	DGP:Bc, DGP:B20, DGP:MBc, DGP:Tc
GUM:Ict	y_t on $1, T$ dummies, & trend	DGP:Bct, DGP:MBct
	Intercept free or forced as noted below; $T = 100$.	

Requirements

Program	`20_02_iis.ox`, `20_02_iis_pcgive.ox`
Software	Ox 7
Dependencies	`20_02_iis.ox`: PcFimlEx and AutometricsExp classes
	`20_02_iis_pcgive.ox`: PcGive and PcGiveExp classes
Also uses	`sim_autometrics.ox`, `sim_pcgive.ox`, `simdesign.ox`, `simdesigns.ox`, `simstore.ox`
Output	`20_02_iis.out` and `20_02_iis_pcgive.out`
Running time	about four hours

`20_02_iis.out` experiments use the PcFimlEx class for the stepwise regression experiments.

Results

$M = 1000$ throughout this chapter.

The DGPs in the first table below have $\delta = 0$, while the constant is free in the GUM, so included in the gauge. The GUM includes IIS, giving table 20.2:

$\alpha = 1\%$	$\gamma = 0$	$\gamma = 1$	$\gamma = 2$	$\gamma = 3$	$\gamma = 4$	$\gamma = 5$
			DGP:Bc, $\delta = 0$			
Gauge %	1.5	1.1	0.9	0.4	0.8	1.0
Potency %	—	6.0	31.1	58.3	89.5	99.2
$\alpha = 1\%$	$\gamma = 0$	$\gamma = 1$	$\gamma = 2$	$\gamma = 3$	$\gamma = 4$	$\gamma = 5$
			DGP:B20, $\delta = 0$			
Gauge %	1.5	1.2	1.0	0.9	1.0	1.0
Potency %	—	4.5	11.8	32.4	73.8	94.8

The DGPs in the next table have $\delta = 1$, while the constant is forced in the GUM, so excluded from the gauge. The GUM includes IIS, giving table 20.3:

	$\gamma = 3$	$\gamma = 4$	$\gamma = 5$	$\gamma = 3$	$\gamma = 4$	$\gamma = 5$
	Autometrics, constant forced, $\alpha = 1\%$					
	DGP:Bc, $\delta = 1$			DGP:Bct, $\delta = 1$		
Gauge %	0.4	0.7	1.1	1.9	1.1	1.2
Potency %	54.8	86.3	99.1	41.5	76.4	97.5
	DGP:MBc, $\delta = 1$			DGP:MBct, $\delta = 1$		
Gauge %	0.5	0.8	1.0	0.5	0.7	1.0
Potency %	35.0	72.5	91.9	38.2	75.4	96.0
	Step-wise regression, constant forced, $\alpha = 1\%$					
	DGP:Bc, $\delta = 1$			DGP:Bct, $\delta = 1$		
Gauge %	0.1	0.1	0.1	0.7	0.4	0.2
Potency %	9.3	12.2	13.7	6.9	6.4	5.8
	DGP:MBc, $\delta = 1$			DGP:MBct, $\delta = 1$		
Gauge %	0.1	0.1	0.1	0.1	0.0	0.2
Potency %	10.0	13.0	14.2	13.7	15.6	18.5

Using PcGive (`20_02_iis_pcgive.ox`), which handles the heteroscedasticity test somewhat differently from PcFiml, makes almost no difference for table 20.2:

$\alpha = 1\%$	$\gamma = 0$	$\gamma = 1$	$\gamma = 2$	$\gamma = 3$	$\gamma = 4$	$\gamma = 5$
	DGP:Bc, $\delta = 0$, PcGive					
Gauge %	1.5	1.1	0.9	0.4	0.8	1.0
Potency %	—	6.0	30.9	58.4	89.5	99.2
$\alpha = 1\%$	$\gamma = 0$	$\gamma = 1$	$\gamma = 2$	$\gamma = 3$	$\gamma = 4$	$\gamma = 5$
	DGP:B20, $\delta = 0$, PcGive					
Gauge %	1.5	1.2	1.0	0.8	1.0	1.0
Potency %	—	4.5	11.8	32.3	73.9	94.8

However, when a trend is included, there is somewhat higher potency with PcGive:

	$\gamma = 3$	$\gamma = 4$	$\gamma = 5$	$\gamma = 3$	$\gamma = 4$	$\gamma = 5$
	Autometrics, constant forced, $\alpha = 1\%$, PcGive					
	DGP:Bc, $\delta = 1$			DGP:Bct, $\delta = 1$		
Gauge %	0.4	0.7	1.1	1.8	1.1	1.2
Potency %	54.8	86.3	99.1	*45.8*	*84.9*	*98.3*
	DGP:MBc, $\delta = 1$			DGP:MBct, $\delta = 1$		
Gauge %	0.5	0.8	1.0	0.5	0.8	1.0
Potency %	35.0	72.5	91.9	38.3	75.3	95.9

20.3 IIS for shifts in the mean of a stationary autoregression

Break with autoregression DGPs

DGP:BLc	$y_t = \delta + \gamma(I_{82} + \cdots + I_{101}) + 0.5y_{t-1} + u_t, \quad y_1 = 0, \quad T = 2, \ldots, 101$
	$u_t \sim \mathsf{IN}[0,1]; \quad \delta = 0, 1$ as noted below

DGP:BL is DGP:BLc($\delta = 0$). We changed the sample dates to reflect the output from the computer program. This does not affect the experiment.

Break with autoregression GUMs

Specification of GUM		Used for DGP
GUM:ILc	y_t on 1 & T dummies & y_{t-1}, t=2,...,101	DGP:BLc
	Intercept free or forced as noted below.	

Requirements

Program	`20_03_iis.ox`
Software	Ox 7
Dependencies	PcFimlEx and AutometricsExp classes
Also uses	`sim_autometrics.ox`, `simdesign.ox`, `simdesigns.ox`, `simstore.ox`
Output	`20_03_iis.out`
Running time	about an hour

The DGP is DGP:BLc with $\delta = 0$ (i.e. DGP:BL) and with $\delta = 1$. The GUM is GUM:ILc.

Results

The results are slightly different from those reported in table 20.4. The reason is that, in the results reported in the book, the first dummy, I_{82} had been omitted. Using all 20 dummies, the experiments give:

	$\gamma = 5$	$\gamma = 8$	$\gamma = 10$	$\gamma = 5$	$\gamma = 8$	$\gamma = 10$
	Autometrics, constant free, $\alpha = 1\%$					
	DGP:BLc, $\delta = 0$			DGP:BLc, $\delta = 1$		
Gauge %	1.4	1.1	1.1	1.6	1.6	1.5
Potency %	43.0	81.5	92.0	13.1	15.7	17.4
	Autometrics, constant forced, $\alpha = 1\%$					
	DGP:BLc, $\delta = 0$			DGP:BLc, $\delta = 1$		
Gauge %	1.4	1.2	1.2	1.4	1.2	1.2
Potency %	46.4	84.7	93.1	46.4	85.5	94.2

20.4 IIS in unit-root models

Break with unit root DGPs

DGP:IUc	y_t	=	$0.2 + \gamma I_{181} + y_{t-1} + u_t,$
DGP:BUc	y_t	=	$0.2 + \gamma (I_{181} + \cdots + I_{200}) + y_{t-1} + u_t,$
DGP:MIUc	y_t	=	$0.2 + \gamma (I_{101} + I_{124} + I_{149} + I_{174} + I_{197}) + y_{t-1} + u_t,$
DGP:MBUc	y_t	=	$0.2 + \gamma (I_{101} + \cdots + I_{104} + I_{124} + \cdots + I_{127} + I_{149} + \cdots$
			$+I_{152} + I_{174} + \cdots + I_{177} + I_{197} + \cdots + I_{200}) + y_{t-1} + u_t,$
			$u_t \sim \mathsf{IN}[0, 1],\ y_1 = 0,\ t = 2, \ldots, 200$

We changed the sample dates to reflect the output from the computer program. This does not affect the experiment.

Break with unit root GUMs

GUM:ILct	y_t on 1, T dummies, y_{t-1}, and trend, $t = 101, \ldots, 200$;
DGUM:ILct	Δy_t on 1, T dummies, y_{t-1}, and trend, $t = 101, \ldots, 200$.

Requirements

Program	`20_04_iis.ox`, `20_04_iis_pcgive.ox`
Software	Ox 7
Dependencies	`20_04_iis.ox`: PcFimlEx and AutometricsExp classes
	`20_04_iis_pcgive.ox`: PcGive and PcGiveExp classes
Also uses	`sim_autometrics.ox`, `sim_pcgive.ox`, `simdesign.ox`, `simdesigns.ox`, `simstore.ox`
Output	`20_04_iis.out`, `20_04_iis_old_2013.out`, `20_04_iis_pcgive.out`
Running time	about seven hours

Results

The results from `20_04_iis_pcgive.ox` are captured in the following table:

	$\gamma = 3$	$\gamma = 4$	$\gamma = 5$	$\gamma = 3$	$\gamma = 4$	$\gamma = 5$
	GUM			DGUM		
	Autometrics, $\alpha = 1\%$, constant forced					
	DGP:IUc			DGP:IUc		
Gauge %	1.4	1.3	1.3	1.6	1.6	1.6
Potency %	83.4	95.6	99.5	83.5	95.6	99.7
	DGP:BUc			DGP:BUc		
Gauge %	1.1	1.0	1.0	1.0	1.1	1.3
Potency %	21.3	39.6	54.8	24.0	60.6	90.8
	DGP:MIUc			DGP:MIUc		
Gauge %	1.0	1.1	1.1	1.4	1.5	1.4
Potency %	55.9	80.0	93.5	66.5	90.0	98.6
	DGP:MBUc			DGP:MBUc		
Gauge %	1.2	2.8	1.9	0.7	0.9	1.0
Potency %	32.9	55.5	62.1	37.8	70.5	93.6

The results in the GUM columns reported above are different from table 20.6 in the book.

The old output file `20_04_iis_old_2013.out` corresponds to table 20.6:

	$\gamma = 3$	$\gamma = 4$	$\gamma = 5$	$\gamma = 3$	$\gamma = 4$	$\gamma = 5$
	GUM			DGUM		
	Autometrics, $\alpha = 1\%$, constant forced					
	DGP:IUc			DGP:IUc		
Gauge %	1.8	1.9	1.8	1.6	1.6	1.6
Potency %	85.1	96.7	99.8	83.5	95.6	99.7
	DGP:BUc			DGP:BUc		
Gauge %	1.5	1.5	1.5	1.0	1.1	1.3
Potency %	21.4	38.0	53.7	24.1	60.6	90.8
	DGP:MIUc			DGP:MIUc		
Gauge %	1.6	1.8	1.9	1.4	1.4	1.4
Potency %	68.6	92.1	99.1	66.5	90.0	98.6
	DGP:MBUc			DGP:MBUc		
Gauge %	1.0	2.6	3.4	0.7	0.9	1.0
Potency %	39.1	71.8	88.0	37.8	70.5	93.6

The old results used PcGive 14.0B3 (part of OxMetrics 7.0) for estimation, together with AutometricsExp. In that version, lagged y_t variables could not be forced (i.e. the U label would be removed). As a consequence, during the search for omitted variables, the lagged dependent variables were not kept fixed, making this search more burdensome. The objective of the algorithm for more variables than observations is to keep the currently maintained model fixed during the search for omitted variables. This is achieved in the current Monte Carlo experiments. There is no impact on the DGUM results, because y_{t-1} is irrelevant there.

20.5 IIS in autoregressions with regressorss

Breaks & regressors DGPs

DGP:Lcx	y_t	$=$	$2 + 0.5y_{t-1} + \boldsymbol{x}'_t\boldsymbol{\beta} + u_t,$
			$y_1 = 0, t = 2, \ldots, 101,$
DGP:BLcx	y_t	$=$	$2 + \gamma\,(I_{82} + \cdots + I_{101}) + 0.5y_{t-1} + \boldsymbol{x}'_t\boldsymbol{\beta} + u_t,$
			$y_1 = 0, t = 2, \ldots, 101,$
DGP:SLcx	y_t	$=$	$2 - \gamma S^*_{81} + 0.5y_{t-1} + \boldsymbol{x}'_t\boldsymbol{\beta} + u_t,$
			$y_0 = 0, t = 2, \ldots, 101,\ S^*_{81} = I_1 + \cdots + I_{81},$
DGP:Ucx	y_t	$=$	$0.2 + y_{t-1} + \boldsymbol{x}'_t\boldsymbol{\beta} + u_t,$
			$y_1 = 0, t = 2, \ldots, 200,$
DGP:BUcx	y_t	$=$	$0.2 + \gamma\,(I_{181} + \cdots + I_{200}) + y_{t-1} + \boldsymbol{x}'_t\boldsymbol{\beta} + u_t,$
			$y_1 = 0, t = 2, \ldots, 200,$
DGP:SUcx	y_t	$=$	$0.2 - \gamma S^*_{180} + y_{t-1} + \boldsymbol{x}'_t\boldsymbol{\beta} + u_t,$
			$y_1 = 0, t = 2, \ldots, 200,\ S^*_{180} = I_1 + \cdots + I_{180}.$

DGP:BLcx and DGP:SLcx differ in that in the former the mean breaks from 2 to $2+\gamma$ at $T = 82$, while in the latter it breaks from $2-\gamma$ to 2 at $T = 82$. In table 20.7, the two versions of DGP:BLcx are the same, because $S_{82} = I_{82} + I_{83} + \cdots$. In earlier versions of SIS, the step dummies were defined like S_{82}, and this is what is used in the book. Subsequently, we decided that it is more useful to define the dummies as S^*_{81}: starting at one and breaking to zero. The advantage of this approach is that only the intercept is extrapolated into the forecast period. In the old approach, the forecasted mean is the sum of all the step-dummy coefficients.

The regressors are created as:

$$\mathbf{x}_t'\boldsymbol{\beta} = \sum_{i=1}^{4} \beta^* \left(x_{i,t} - x_{i,t-1}\right) \tag{6}$$

where:

$$\begin{aligned} x_{i,t} &= \rho x_{i,t-1} + v_{i,t}, \quad i = 1, \ldots, 10; \\ v_{i,t} &\sim \mathsf{IN}\left[0, (1-\rho)^2\right]; \\ \beta &= \{2.4, 3.2, 4.0\}. \end{aligned}$$

Breaks & regressors GUM

GUM:Lcx	y_t on 1, y_{t-1}, X.
GUM:ILcx	y_t on 1, T dummies, y_{t-1}, X.
GUM:SLcx	y_t on 1, T dummies, $T-1$ levels S^*, y_{t-1}, X.
GUM:Lcx	y_t on 1, y_{t-1}.
GUM:Lcxt	y_t on 1, y_{t-1}, and trend.
GUM:ILcxt	y_t on 1, T dummies, trend, y_{t-1}, X.
GUM:SLcxt	y_t on 1, T dummies, $T-1$ levels S^*, trend, y_{t-1}, X.

All estimated models use the last 100 observations of the generated samples.

$$\text{where } X = \sum_{i=1}^{10} \left(\gamma_i x_{i,t} + \delta_i x_{i,t-1}\right).$$

Requirements

Program	`20_05_areg.ox`, `20_05_iis.ox`, `20_05_iis_pcgive.ox`, `20_05_iis_sis.ox`
Software	Ox 7
Dependencies	`20_04_iis.ox`, `20_05_iis_sis.ox`: PcFimlEx and AutometricsExp classes
	`20_05_areg.ox`, `20_04_iis_pcgive.ox`: PcGive and PcGiveExp classes
Also uses	`sim_autometrics.ox`, `simdesign.ox`, `simdesigns.ox`, `simstore.ox`
Output	`20_05_areg.out`, `20_05_iis.out`, `20_05_iis_pcgive.out`, `20_05_iis_sis.out`
	`20_05_iis_old_2013.out`
Running time	about 15 minutes for `20_05_areg.ox`, from one to three hours for the others

Results

The first table, 20.8 in the book, is obtained from `20_05_areg.ox`:

	$\rho = 0$			$\rho = 0.9$		
	$\beta = 2.4$	$\beta = 3.2$	$\beta = 4.0$	$\beta = 2.4$	$\beta = 3.2$	$\beta = 4.0$
***Autometrics*, Constant free,** $\alpha = 1\%$						
	DGP:Lcx, GUM:Lcx			DGP:Lcx, GUM:Lcx		
Gauge %	2.4	2.7	2.2	3.9	4.1	3.4
Potency %	46.0	65.0	82.6	40.0	59.5	80.0
	DGP:Ucx, GUM:Lcx			DGP:Ucx, GUM:Lcx		
Gauge %	2.8	2.6	1.9	3.4	3.5	2.9
Potency %	46.9	67.7	84.9	36.4	56.0	75.5
	DGP:Ucx, GUM:Lcxt			DGP:Ucx, GUM:Lcxt		
Gauge %	4.8	4.8	4.1	10.8	11.7	10.8
Potency %	42.9	62.7	79.5	37.8	55.1	72.0
***Autometrics*, Constant forced,** $\alpha = 1\%$						
	DGP:Lcx, GUM:Lcx			DGP:Lcx, GUM:Lcx		
Gauge %	2.4	2.7	2.2	3.9	4.1	3.4
Potency %	41.0	61.8	81.0	34.6	55.8	78.2
	DGP:Ucx, GUM:Lcx			DGP:Ucx, GUM:Lcx		
Gauge %	2.7	2.5	1.9	2.5	2.8	2.0
Potency %	43.4	66.3	85.2	33.2	56.2	78.6
	DGP:Ucx, GUM:Lcxt			DGP:Ucx, GUM:Lcxt		
Gauge %	5.5	5.6	4.8	11.2	12.3	11.3
Potency %	42.1	63.7	83.0	34.1	53.3	71.8

The second table is obtained from `20_05_iis.ox` and `20_05_iis_sis.ox`. This is different from table 20.9 for the reasons given in the previous section.

	$\rho = 0$		
	$\beta = 2.4$	$\beta = 3.2$	$\beta = 4.0$
	Autometrics, Constant forced, $\alpha = 1\%$		
	DGP:BLcx, GUM:ILcx, $\gamma = 10$		
Gauge %	2.1	2.1	1.9
Potency %	58.8	61.0	66.3
	DGP:SLcx, GUM:SLcx, $\gamma = 10$		
Gauge %	1.6	1.8	1.7
Potency %	44.3	62.7	79.9
	DGP:BUcx, GUM:ILcxt, $\gamma = 5$		
Gauge %	1.6	1.6	1.5
Potency %	36.2	39.1	43.9
	DGP:SUcx,GUM:SLcxt, $\gamma = 5$		
Gauge %	1.8	1.9	1.8
Potency %	43.8	62.5	79.1

22 Testing Super Exogeneity

22.9 Testing exogeneity in DHSY

Requirements

Program	`22_09_DHSY.fl`
Data	`DHSY.xlsx`
Software	PcGive
Output	`22_09_DHSY.out`

Results

`22_09_DHSY.fl` estimates the reduced form equations for y_t and $\Delta\Delta_4 p_t$, finding seven dummies for the former, and five for the latter:

$\Delta_4 c_t$ (16.5)	y	$\Delta\Delta_4 p_t$
I:1962(2)		
	I:1959(2)	I:1959(2)
	I:1966(1)	
	I:1966(2)	
	I:1968(2)	
I:1972(1)		
		I:1972(2)
		I:1974(1)
	I:1974(2)	I:1974(2)
	I:1974(3)	
	I:1975(2)	I:1975(2)

Adding the nine dummies to (16.5), which already contains I:1962(2) and I:1972(1), yields:

```
EQ(90) Modelling D4c by OLS
       The dataset is: D:\Documents\Books_other\Lund\code\DHSY.xlsx
       The estimation sample is: 1959(1) - 1975(4)

                  Coefficient  Std.Error  t-value  t-prob Part.R^2
I:1962(2)           0.0159207   0.005822     2.73  0.0088   0.1373
I:1972(1)           0.0118270   0.005804     2.04  0.0472   0.0812
I:1959(2)          0.00503427   0.006180    0.815  0.4194   0.0139
I:1966(1)         -0.00790529   0.006167    -1.28  0.2061   0.0338
I:1966(2)          0.00177537   0.006303    0.282  0.7794   0.0017
I:1968(2)        -0.000965778   0.006442   -0.150  0.8815   0.0005
I:1974(2)          0.00431331   0.007432    0.580  0.5645   0.0071
I:1974(3)          0.00386877   0.006750    0.573  0.5693   0.0069
I:1975(2)          0.00464265   0.006844    0.678  0.5009   0.0097
I:1972(2)         -0.00808978   0.006523    -1.24  0.2211   0.0317
I:1974(1)         -0.00398310   0.006680   -0.596  0.5539   0.0075
```

```
(continued)        Coefficient  Std.Error  t-value  t-prob Part.R^2
Constant      U   -0.00688718   0.005430    -1.27  0.2109   0.0331
D4y           U      0.486890    0.03134     15.5  0.0000   0.8370
DD4y          U     -0.200990    0.04550    -4.42  0.0001   0.2934
D4p           U     -0.169757    0.03143    -5.40  0.0000   0.3830
DD4p          U     -0.239846     0.1261    -1.90  0.0633   0.0715
c-y_4         U     -0.149896    0.03948    -3.80  0.0004   0.2347
CSeasonal     U   -0.00972626   0.002981    -3.26  0.0021   0.1846
CSeasonal_1   U   -0.00650070   0.002588    -2.51  0.0155   0.1183
CSeasonal_2   U   -0.00401522   0.002075    -1.93  0.0591   0.0738
D4D           U    0.00532327   0.002579     2.06  0.0446   0.0831

sigma               0.00543668  RSS             0.00138920347
R^2                   0.911923  F(20,47) =    24.33 [0.000]**
Adj.R^2               0.874443  log-likelihood        270.662
no. of observations         68  no. of parameters          21
mean(D4c)            0.0232562  se(D4c)             0.0153431

AR 1-5 test:      F(5,42)    =   1.0068 [0.4257]
ARCH 1-4 test:    F(4,60)    =  0.46650 [0.7600]
Normality test:   Chi^2(2)   =  0.76215 [0.6831]
Hetero test:      F(15,41)   =  0.54362 [0.8992]
Hetero-X test: not enough observations
RESET23 test:     F(2,45)    =   1.6461 [0.2042]

Test for excluding:
[0] = I:1959(2)
[1] = I:1966(1)
[2] = I:1966(2)
[3] = I:1968(2)
[4] = I:1974(2)
[5] = I:1974(3)
[6] = I:1975(2)
[7] = I:1972(2)
[8] = I:1974(1)
Subset F(9,47)   =  0.60100 [0.7896]
```

22.10 IIS and economic interpretations

Requirements

Program `22_10_DHSY_VAR_forc.fl`
Data `DHSY.xlsx`
Software PcGive
Output `22_10_DHSY_VAR_forc.out`

Results

`22_10_DHSY_VAR_forc.fl` generates 1-step forecasts from the VAR(1) with trend and seasonals, see Figure 17 and Figure 18.

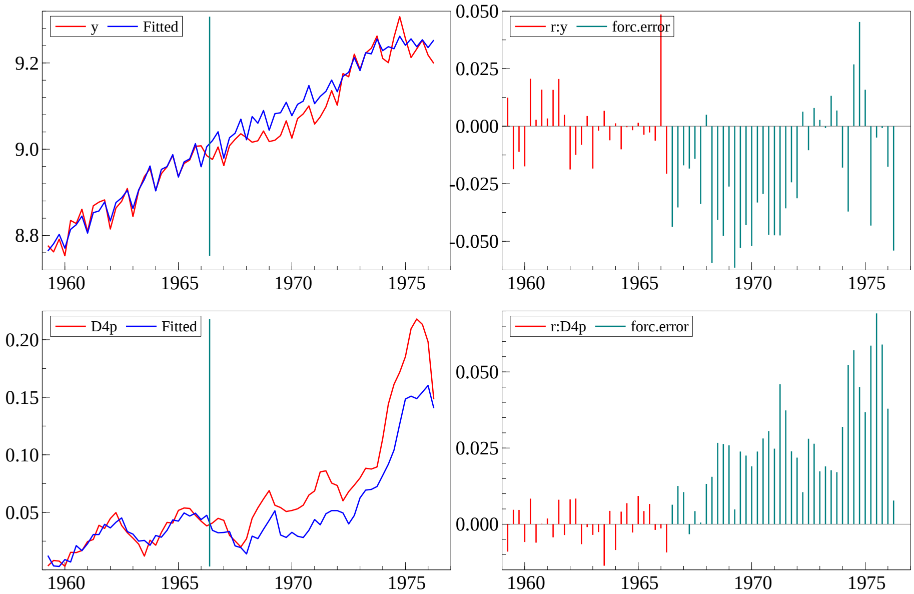

Figure 17
1-step forecasts from VAR(1) with forecast errors

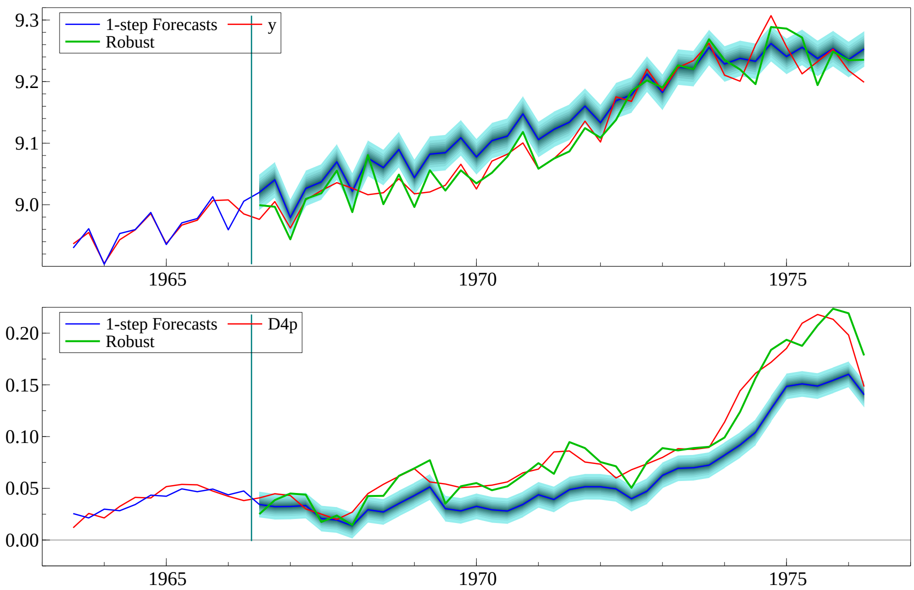

Figure 18
1-step forecasts from VAR(1) with robust forecast

23 Selecting Forecasting Models

23.11 Some simulation findings

DGP

There are 7 experiments, indexed by k, each of which has the same noncentrality for all relevant variables:

$$\begin{aligned} y_t &= \textstyle\sum_{i=1}^{5} \beta_k z_{i,t} + \epsilon_t, \quad \epsilon_t \sim \mathsf{IN}\left[0, 1\right], \\ z_t &= (z_{1,t}, \ldots, z_{10,t})', \quad z_t \sim \mathsf{IN}_{10}\left[\mathbf{0}, \boldsymbol{I}_{10}\right]. \end{aligned}$$

DGP coefficients

	$k = 1$	2	3	4	5	6	7
β_k	$0.1/T^{1/2}$	$1/T^{1/2}$	$2/T^{1/2}$	$3/T^{1/2}$	$4/T^{1/2}$	$5/T^{1/2}$	$8/T^{1/2}$

GUM

$$y_t = \gamma_0^F + \textstyle\sum_{i=1}^{10} \gamma_i z_{i,t} + u_t. \tag{7}$$

So there are five irrelevant variable, and the non-centralities in the experiments are respectively: $0.1, 2, 3, 4, 5, 8$. $T = 100$, $M = 1000$, z_t not fixed.

Requirements

Program	`23_11_forecasting.ox`
Software	Ox 7
Dependencies	PcFimlEx and Autometrics classes
Also uses	`simdesign.ox`, `simstore.ox`, `simutils.ox`
Output	`23_11_forecasting.out`, `23_11_msfe_*.in7/.bn7`
Running time	about 45 minutes (only a few minutes without all possible models)

Results

Figure 19a compares mean squared forecast errors from the 1-step ahead forecast (for $t = 101$) from the DGP, the GUM and *Autometrics* selection at 5%:

Autometrics 5% the final terminal model from *Autometrics*, $\alpha = 5\%$;

Autometrics 5% BC2 the final terminal model from *Autometrics*, $\alpha = 5\%$, using the two-step bias correction;

MA(AutT,U,SC) the average over all terminal model from *Autometrics*, $\alpha = 5\%$, using BIC for the weights.

In this case, there is a small improvement from averaging over terminal models.[1] There is some gain from bias correction up to $t = 2.5$, and a comparable worsening up to t-values of five.

Example of averaging over two models:

Models		
DGP	Model 1 estimates	Model 2 estimates
β_1	$\widehat{\beta}_{11}$	$\widehat{\beta}_{12}$
β_2	$\widehat{\beta}_{21}$	0
weights		
BIC	ζ_1	ζ_2
BIC weights	$w_1 = e^{-\zeta_1/2}$	$w_2 = e^{-\zeta_2/2}$
equal weights	$w_1 = 1/2$	$w_2 = 1/2$
Model averages		
	Unconditional model 1	Conditional model 2
	$\widetilde{\beta}_1^U = (w_1\widehat{\beta}_{11} + w_2\widehat{\beta}_{12})/(w_1 + w_2)$	$\widetilde{\beta}_1^C = (w_1\widehat{\beta}_{11} + w_2\widehat{\beta}_{12})/(w_1 + w_2)$
	$\widetilde{\beta}_2^U = (w_1\widehat{\beta}_{21})/(w_1 + w_2)$	$\widetilde{\beta}_2^C = (w_1\widehat{\beta}_{21})/w_1 = \widehat{\beta}_{21}$

Note that in these settings model averaging and forecast averaging are the same.

Figure 19b compares mean squared forecast errors from the 1-step ahead forecast (for $t = 101$) from the DGP, the GUM and:

Autometrics 5% the final terminal model from *Autometrics*, $\alpha = 5\%$;

MA(2^k,U,SC) the unconditional average over all possible model, using BIC for the weights;

MA(2^k,C,SC) the conditional average over all possible model, using BIC for the weights.

Averaging over all possible models takes the GUM (7) as the starting point, using all possible subsets of the free regressors: in this case there are $2^{10} = 1024$ models, all including an intercept. Unconditional averaging over all possible models acts as a coefficient shrinkage.[2] This is very bad when there is a large non-centrality, but beneficial for small non-centralities. Conditional averaging is very similar to forecasting from the GUM, except that it gets a bit worse when the non-centralities are large.

The Figure 19a results using BC2 are comparable to those reported in the book as figure 23.2. They are not exactly identical because the implementation of the bias correction has changed somewhat from the original results.

Figure 20 shows the impact of α on model selection using *Autometrics*.

[1]There is almost no difference between using BIC weights and equal weights.

[2]With equal weights, because a regressor is included in half of the models, it amounts to dividing the average estimated coefficient by two.

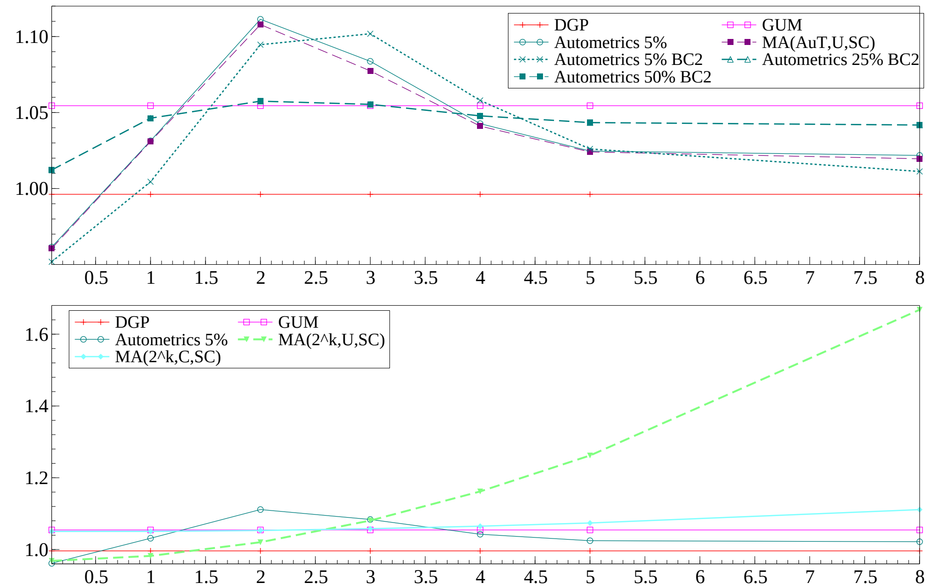

Figure 19
MSFE of 1-step ahead forecast from different methods

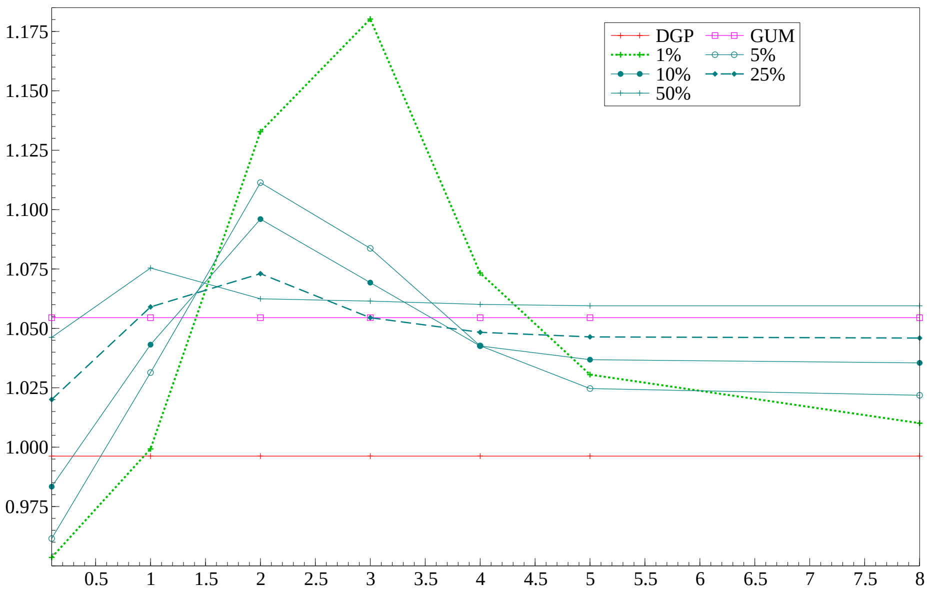

Figure 20
MSFE of 1-step ahead forecast from *Autometrics* using different values of the significance level α

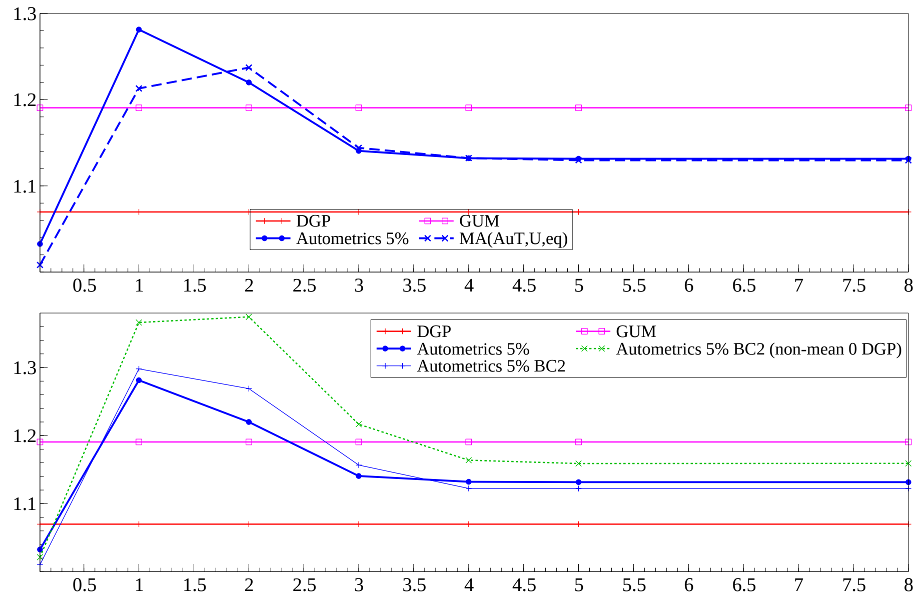

Figure 21
MSFE of 1-step ahead forecast using integrated regressors, all but one mean-zero DGP

Treatment of the intercept matters

The DGP used sofar in the forecasting exercise has normal variables, all with expectation zero. To change this, the generation of the regressors is altered:

$$\begin{aligned} z_t &= (z_{1,t}, \ldots, z_{10,t})', \quad z_t \sim \mathsf{IN}_{10}\left[\mathbf{0}, \boldsymbol{I}_{10}\right], \\ z^s_{i,t} &= z_{i,t} + z^s_{i,t-1}, \qquad z^s_{i,-1} = 0. \end{aligned}$$

The modified DGP uses z^s_t, which are the cumulated regressors. A third, mean-zero version is used, where the integrated regressors are given a zero mean over the entire sample (including the forecast period).

Figure 21a shows that the profile of MSFE from *Autometrics* using integrated regressors is similar to that before, although improvement over the GUM is now reached at a t-value of 2.5 rather than 1.5.

Figure 21b shows that bias correction makes the MSFE *Autometrics* worse, using the non-standardized DGP. The reason is that bias correction leaves the intercept, which is forced, unadjusted, so that the resulting estimated intercept is worse. This is confirmed by the results from the mean 0 DGP, where the impact of bias correction is smaller.

Figure 22 shows the effect of model averaging when the regressors are I(1).

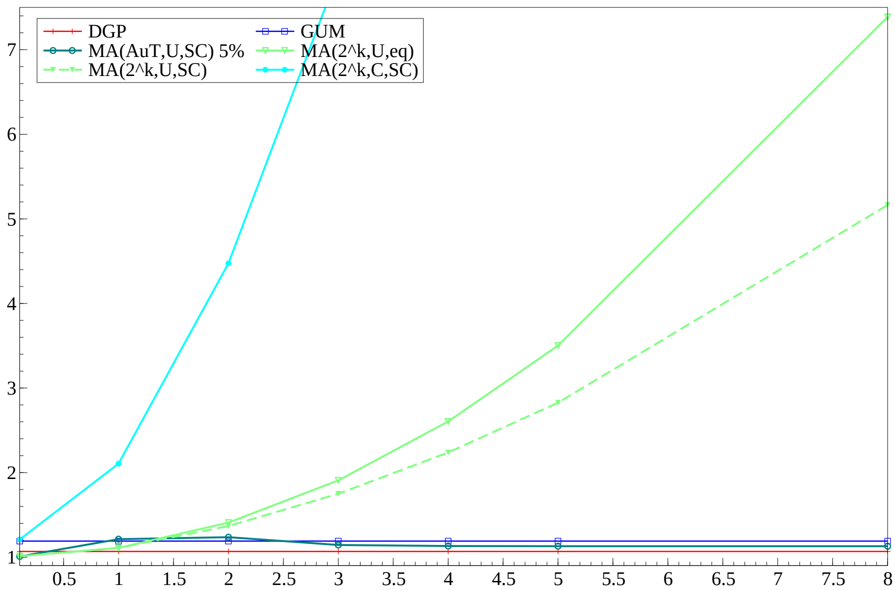

Figure 22
MSFE of 1-step ahead forecast using integrated regressors with mean zero

References

Castle, J. L. (2005). Evaluating PcGets and RETINA as automatic model selection algorithms. *Oxford Bulletin of Economics and Statistics 67*, 837–880. [38, 39]

Davidson, J. E. H., D. F. Hendry, F. Srba, and J. S. Yeo (1978). Econometric modelling of the aggregate time-series relationship between consumers' expenditure and income in the United Kingdom. *Economic Journal 88*, 661–692. Reprinted in Hendry, D. F., *Econometrics: Alchemy or Science?* Oxford: Blackwell Publishers, 1993, and Oxford University Press, 2000; and in Campos, J., Ericsson, N.R. and Hendry, D.F. (eds.), *General to Specific Modelling*. Edward Elgar, 2005. [viii, 29]

Doornik, J. A. (2009). Autometrics. In J. L. Castle and N. Shephard (Eds.), *The Methodology and Practice of Econometrics: Festschrift in Honour of David F. Hendry*. Oxford: Oxford University Press. [33, 40]

Doornik, J. A. (2013). *Object-Oriented Matrix Programming using Ox* (7th ed.). London: Timberlake Consultants Press. [vii]

Ericsson, N. R., D. F. Hendry, and K. M. Prestwich (1998). The demand for broad money in the United Kingdom, 1878–1993. *Scandinavian Journal of Economics 100*, 289–324. [viii]

Friedman, M. and A. J. Schwartz (1982). *Monetary Trends in the United States and the United Kingdom: Their Relation to Income, Prices, and Interest Rates, 1867–1975*. Chicago: University of Chicago Press. [viii]

Hendry, D. F. (1999). An econometric analysis of US food expenditure, 1931–1989. See Magnus and Morgan (1999), pp. 341–361. [viii]

Hendry, D. F. (2009). The methodology of empirical econometric modeling: Applied econometrics through the looking-glass. In T. C. Mills and K. D. Patterson (Eds.), *Palgrave Handbook of Econometrics*, pp. 3–67. Basingstoke: Palgrave MacMillan. [viii]

Hendry, D. F. and J. A. Doornik (2013). *Empirical Econometric Modelling using PcGive: Volume I* (7th ed.). London: Timberlake Consultants Press. [vii]

Hoover, K. D. and S. J. Perez (1999). Data mining reconsidered: Encompassing and the general-to-specific approach to specification search. *Econometrics Journal 2*, 167–191. [viii]

Magnus, J. R. and M. S. Morgan (Eds.) (1999). *Methodology and Tacit Knowledge: Two Experiments in Econometrics*. Chichester: John Wiley and Sons. [viii, 65]

Reade, J. J. (2008). Updating Tobin's food expenditure time series data. Working paper, Department of Economics, University of Oxford. [viii]

Tobin, J. (1950). A statistical demand function for food in the U.S.A. *Journal of the Royal Statistical Society A, 113*(2), 113–141. [viii]