ETH zürich

KOF

Methods of Macroeconomic Forecasting

SEM - Lab 2

KOF ETH Zurich
October 3, 2025, Zurich

SEM

Basic steps

1. Construct a string vector containing all the equations of the model:
   - Decide whether equations are stochastic or an identity.
   - Declare all exogenous variables.
   - Choose informative priors or use non-informative priors (default).
2. Load or construct the data set to match the variable names in the string vector.
3. Estimate the model using the `estimate` function.
4. Compute forecasts using the `forecast` function.
5. Print estimates using `print` and `summary`.
6. Plot or print forecasts using `plot` or `print`.

Example: A Small Macroeconomic Model

- Consider the following model

$$\begin{aligned}
\text{private consumption:} \quad & c_t = \alpha_c + \beta_{c1} d_t + \beta_{c2} c_{t-1} + \varepsilon_{t1} \\
\text{investment:} \quad & i_t = \alpha_i + \beta_{i1} r_t + \beta_{i2} i_{t-1} + \varepsilon_{t2} \\
\text{interest rate:} \quad & r_t = \alpha_r + \beta_{r1} r_t^w + \beta_{r2} r_{t-1} + \varepsilon_{t3} \\
\text{private consumption deflator:} \quad & p_t^c = \alpha_{pc} + \beta_{pc1} fx_t + \beta_{pc2} p_t^{oil} + \beta_{pc3} p_{t-1}^c + \varepsilon_{t4} \\
\text{investment deflator:} \quad & p_t^i = \alpha_{pi} + \beta_{pi1} p_{t-1}^i + \varepsilon_{t5} \\
\text{domestic demand:} \quad & d_t = \omega_c c_t + \omega_g g_t + \omega_i i_t
\end{aligned}$$

  - c_t, i_t, r_t, p_t and d_t are endogenous variables
  - g_t, p_t^w and r_t^w are exogenous
  - c_{t-1} is predetermined
  - d_t is defined by an identity equation

KOMA Implementation

Define the following string vector

consumption:	`"consp ~ domdemoi + consp.L(1),`
investment:	`ifix ~ srate + ifix.L(1),`
interest rate:	`srate ~ constant + srate_ge + srate.L(1),`
consumption deflator:	`pconsp ~ wkfreuro + poilusd + pconsp.L(1),`
investment deflator:	`pifix ~ pifix.L(1),`
demand:	`domdemoi == (nconsp/ndomdemoi)*consp`
	`+ (nconsg/ndomdemoi)*consg`
	`+ (nifix/ndomdemoi)*ifix"`

KOMA Implementation

Define the following string vector

consumption:	`"consp ∼ domdemoi + consp.L(1),`
investment:	`ifix ∼ srate + ifix.L(1),`
interest rate:	`srate ∼ constant + srate_ge + srate.L(1),`
consumption deflator:	`pconsp ∼ wkfreuro + poilusd + pconsp.L(1),`
investment deflator:	`pifix ∼ pifix.L(1),`
demand:	`domdemoi == (nconsp/ndomdemoi)*consp`
	`+ (nconsg/ndomdemoi)*consg`
	`+ (nifix/ndomdemoi)*ifix"`

KOMA Implementation

Define the following string vector

consumption:	`"consp ~ 0 + domdemoi + consp.L(1),`
investment:	`ifix ~ srate + ifix.L(1),`
interest rate:	`srate ~ constant + srate_ge + srate.L(1),`
consumption deflator:	`pconsp ~ wkfreuro + poilusd + pconsp.L(1),`
investment deflator:	`pifix ~ pifix.L(1),`
demand:	`domdemoi == (nconsp/ndomdemoi)*consp`
	`+ (nconsg/ndomdemoi)*consg`
	`+ (nifix/ndomdemoi)*ifix"`

KOMA Implementation

Define the following string vector

consumption:	`"consp ∼ domdemoi + consp.L(1),`
investment:	`ifix ∼ srate + ifix.L(1),`
interest rate:	`srate ∼ constant + srate_ge + srate.L(1),`
consumption deflator:	`pconsp ∼ wkfreuro + poilusd + pconsp.L(1),`
investment deflator:	`pifix ∼ pifix.L(1),`
demand:	`domdemoi == (nconsp/ndomdemoi)*consp`
	`+ (nconsg/ndomdemoi)*consg`
	`+ (nifix/ndomdemoi)*ifix"`

KOMA Implementation

Define the following string vector

consumption:	`"consp ~ domdemoi + consp.L(1),`
investment:	`ifix ~ srate + ifix.L(1),`
interest rate:	`srate ~ constant + srate_ge + srate.L(1),`
consumption deflator:	`pconsp ~ wkfreuro + poilusd + pconsp.L(1),`
investment deflator:	`pifix ~ pifix.L(1),`
demand:	`domdemoi == (nconsp/ndomdemoi)*consp`
	`+ (nconsg/ndomdemoi)*consg`
	`+ (nifix/ndomdemoi)*ifix"`

KOMA Implementation

Define the following string vector

consumption:	`"consp ∼ domdemoi + consp.L(1),`
investment:	`ifix ∼ srate + ifix.L(1),`
interest rate:	`srate ∼ constant + srate_ge + srate.L(1),`
consumption deflator:	`pconsp ∼ wkfreuro + poilusd + pconsp.L(1),`
investment deflator:	`pifix ∼ pifix.L(1),`
demand:	`domdemoi == (nconsp/ndomdemoi)*consp`
	`+ (nconsg/ndomdemoi)*consg`
	`+ (nifix/ndomdemoi)*ifix"`

KOMA Implementation

Define the following string vector

consumption:	`"consp ~ domdemoi + lag(consp,1),`
investment:	`ifix ~ srate + ifix.L(1),`
interest rate:	`srate ~ constant + srate_ge + srate.L(1),`
consumption deflator:	`pconsp ~ wkfreuro + poilusd + pconsp.L(1),`
investment deflator:	`pifix ~ pifix.L(1),`
demand:	`domdemoi == (nconsp/ndomdemoi)*consp`
	`+ (nconsg/ndomdemoi)*consg`
	`+ (nifix/ndomdemoi)*ifix"`

KOMA Implementation

Define the following string vector

consumption:	`"consp ∼ domdemoi + consp.L(1),`
investment:	`ifix ∼ srate + ifix.L(1),`
interest rate:	`srate ∼ constant + srate_ge + srate.L(1),`
consumption deflator:	`pconsp ∼ wkfreuro + poilusd + pconsp.L(1),`
investment deflator:	`pifix ∼ pifix.L(1),`

demand:	`domdemoi == (nconsp/ndomdemoi)*consp`
	`+ (nconsg/ndomdemoi)*consg`
	`+ (nifix/ndomdemoi)*ifix"`

KOMA Implementation

Define the following string vector

consumption:	`"consp ~ domdemoi + consp.L(1),`
investment:	`ifix ~ srate + ifix.L(1),`
interest rate:	`srate ~ constant + srate_ge + srate.L(1),`
consumption deflator:	`pconsp ~ wkfreuro + poilusd + pconsp.L(1),`
investment deflator:	`pifix ~ pifix.L(1),`
demand:	`domdemoi == (nconsp/ndomdemoi)*consp`
	`+ (nconsg/ndomdemoi)*consg`
	`+ (nifix/ndomdemoi)*ifix"`

KOMA Implementation

Define the following string vector

consumption:	`"consp ~ domdemoi + consp.L(1),`
investment:	`ifix ~ srate + {mu,sigma}ifix.L(1),`
interest rate:	`srate ~ constant + srate_ge + srate.L(1),`
consumption deflator:	`pconsp ~ wkfreuro + poilusd + pconsp.L(1),`
investment deflator:	`pifix ~ pifix.L(1),`
demand:	`domdemoi == (nconsp/ndomdemoi)*consp`
	`+ (nconsg/ndomdemoi)*consg`
	`+ (nifix/ndomdemoi)*ifix"`

KOMA Implementation

Define the following string vector

...

```
...
```

demand:

```
domdemoi == (nconsp/ndomdemoi)*consp
+ (nconsg/ndomdemoi)*consg
+ (nifix/ndomdemoi)*ifix"
```

nominal consumption:

```
nconsp == 1*consp+ 1*pconsp
```

nominal investment:

```
nifix == 1*ifix + 1*pifix
```

Hands-on

Use data from `koma::small_open_economy`.

```
consumption ~ gdp + consumption.L(1)
investment ~ investment.L(1)
exports ~ world_gdp + exports.L(1)
imports ~ domestic_demand + imports.L(1)
inflation ~ exchange_rate + oil_price + inflation.L(1)
interest_rate ~ inflation + interest_rate_germany + inflation.L(1)
gdp == 0.6*consumption + 0.6*domestic_demand
domestic_demand == 0.6*consumption + 0.4*investment
```