gets: An R Package for General-to-Specific (GETS) Modelling and Indicator Saturation Methods

Genaro Sucarrat

Department of Economics
BI Norwegian Business School

http://www.sucarrat.net/

Aalborg, 2 July 2015
The \( R \) package \texttt{gets} provides automated...

1. ...GETS modelling of the mean of a regression:

\[
y_t = \beta_1 x_{1t} + \cdots + \beta_k x_{kt} + \epsilon_t, \quad \epsilon_t = \sigma_t z_t, \quad z_t \sim iid(0, 1)
\]

2. ...GETS modelling of the (log-)variance of a regression:

\[
\ln \sigma_t^2 = \alpha_1 w_{1t} + \cdots + \alpha_l w_{lt}
\]

3. ...detection and tests for breaks in the mean-intercept (w/Indicator Saturation (IS) methods)

The main functions of the \texttt{gets} package:

1. \texttt{arx}: Estimation of AR-X model with (optionally) log-ARCH-X errors
2. \texttt{getsm}: Automated GETS modelling of mean specification
3. \texttt{getsv}: Automated GETS modelling of variance specification
4. \texttt{isat}: Automated GETS modelling of an indicator saturated mean specification
Automated multi-path GETS modelling software:


- **Sucarrat (October 2014):** gets. An R package available on the CRAN. More user-friendly and faster than AutoSEARCH, and contains more features (e.g. indicator saturation methods)

- **Hoover and Perez (1999):** MATLAB code. Only 10 paths, not user-friendly and no help-system

- **OxMetrics (commercial):**
Why AutoSEARCH/gets?:

- **PcGets/Autometrics** models the mean:
  \[ y_t = \phi_0 + \sum_r \phi_r y_{t-r} + \sum_s \eta_s x_{s,t}^m + \epsilon_t, \quad \epsilon_t = \sigma_t z_t, \quad z_t \sim iid(0, 1) \]

- In my research, I was interested in GETS modelling of the log-variance:
  \[ \ln \sigma_t^2 = \alpha_0 + \sum_p \alpha_p \ln \epsilon_{t-p}^2 + \sum_d \delta_d x_{d,t}^\nu \]

- **PcGets/Autometrics** achieves this by modelling the AR-X representation:
  \[ \ln \epsilon_t^2 = \phi_0 + \sum_{p=1}^P \alpha_p \ln \epsilon_{t-p}^2 + \sum_{d=1}^D \delta_d x_{d,t}^\nu + u_t, \quad u_t \sim iid(0, \sigma_u^2), \]

where \( \phi_0 = \alpha_0 + E(\ln z_t^2) \) and \( u_t = \ln z_t^2 - E(\ln z_t^2) \), see e.g. Bauwens and Sucarrat (2010): “General to Specific Modelling of Exchange Rate Volatility: A Forecast Evaluation”, *Int.J.Forecasting* 26, pp. 885-907

- Problems: Uncorrelated and homoscedastic residuals \( \widehat{u}_t \) does not imply uncorrelated and homoscedastic standardised residuals \( \widehat{z}_t \), and likelihood-based comparisons with other models should preferrably be undertaken in terms of the likelihood of \( \widehat{\epsilon}_t \) rather than of \( \widehat{u}_t \)

Does anyone use AutoSEARCH/gets?
Main ingredients of GETS modelling:

- Backwards elimination (along multiple paths)
- Regressor significance testing (individual and joint)
- Diagnostics testing
- Information criteria

GETS modelling in 3 steps:

1: Formulate a General Unrestricted Model (GUM) that passes the chosen diagnostics tests

2: Backwards elimination of insignificant regressors along multiple paths while at each regressor removal: a) Test for joint insignificance and b) Check the diagnostics

3: Choose the best terminal model according to an information criterion
Model selection properties of GETS modelling:

- $k_0$: Number of relevant variables
- $k_1$: Number of irrelevant variables
- $E(\hat{k}_1/k_1) \to \alpha$: The irrelevance proportion or “gauge” should equal the significance level $\alpha$
- $E(\hat{k}_0/k_0) \to 1$: The relevance proportion or “potency”
- The (L)DGP is contained in the final model with probability 1
- The irrelevance proportion (i.e. gauge) is closely related to the Per Comparison Error Rate (PCER)
- The gauge and potency can be viewed as a more detailed (and arguably more intuitive) characterisation of the False Discovery Rate (FDR)

How well does GETS modelling compared with other model selection algorithms?

- Very well!
- Studies show that it generally does better than step-wise methods
- Studies show that it generally beats the LASSO/shrinkage methods
Selected reading:


- Hendry, Johansen and Santos Hendry et al. (2007): “Automatic selection of indicators in a fully saturated regression”, *Computational Statistics*


Outline:

- **arx**: Estimation of AR-X model with (optionally) log-ARCH-X errors
- **getsm**: GETS modelling of mean specification
- **getsv**: GETS modelling of (log)variance specification
- **isat**: GETS modelling of an indicator saturated mean specification
- **Future versions**: Vignette, further speed improvements, additional features
arx: Estimation
The AR-X model with log-ARCH-X errors is given by
\begin{align*}
y_t &= \phi_0 + \sum_r \phi_r y_{t-r} + \sum_s \eta_s x_{s,t}^m + \epsilon_t, \quad \epsilon_t = \sigma_t z_t, \quad z_t \sim iid(0,1) \\
\ln \sigma_t^2 &= \alpha_0 + \sum_p \alpha_p \ln \epsilon_{t-p}^2 + \sum_d \delta_d x_{d,t}^v
\end{align*}

Example of \texttt{arx}:

```r
set.seed(123)
y <- arima.sim(list(ar=0.4), 100)
mod01 <- arx(y, ar=1)
```

Let us make things more interesting...

```r
mX <- matrix(rnorm(100*5), 100, 5)
mod02 <- arx(y, mc=TRUE, ar=1:2, mxreg=mX)
mod03 <- arx(y, mc=TRUE, ar=1:2, mxreg=mX, arch=1:3, asym=1, vxreg=log(mX^2), vcov.type="white")
```

Extraction functions:

```
coef, fitted, plot, print, residuals, summary, vcov
```
getsmp: Modelling the mean
Usage of getsm: Apply on arx object

Examples:

```r
getsm02 <- getsm(mod02)
getsm02b <- getsm(mod02, t.pval=0.01, wald.pval=0.01)
getsm02c <- getsm(mod02, keep=1)
```

All arguments of getsm function (w/defaults):

- `keep = NULL`, `vcov.type = NULL`, `t.pval = 0.05`, `do.pet = TRUE`, `wald.pval = 0.05`, `ar.LjungB = list(lag = NULL, pval = 0.025)`, `arch.LjungB = list(lag = NULL, pval = 0.025)`, `info.method = c("sc", "aic", "hq"), include.empty = FALSE`, `zero.adj = NULL`, `vc.adj = NULL`, `tol = NULL`, `LAPACK = NULL`, `max.regs = 1e+05`, `verbose = TRUE`, `print.searchinfo = TRUE`, `alarm = FALSE`

Extraction functions:

- `coef`, `fitted`, `paths`, `plot`, `print`, `residuals`, `summary`, `terminals`, `vcov`
**getsv**: Modelling the variance
Usage of `getsv`: Apply on `arx` object

Examples:

```r
getsv03 <- getsv(mod03)
getsv03b <- getsv(mod03, t.pval=0.1, wald.pval=0.1)
getsv03c <- getsv(mod03, keep=1:4)
```

All arguments of `getsv` function (w/defaults):

- `keep = c(1)`, `t.pval = 0.05`, `do.pet = TRUE`, `wald.pval = 0.05`, `ar.LjungB = list(lag = NULL, pval = 0.025)`, `arch.LjungB = list(lag = NULL, pval = 0.025)`, `info.method = c("sc", "aic", "hq")`, `include.empty = FALSE`, `zero.adj = NULL`, `vc.adj = NULL`, `tol = NULL`, `LAPACK = NULL`, `max.regs = 1e+05`, `verbose = TRUE`, `alarm = FALSE`

Extraction functions (same as those of `getsm`):

- `coef`, `fitted`, `paths`, `plot`, `print`, `residuals`, `summary`, `terminals`, `vcov`
isat: Indicator Saturation
isat function:

- Joint with Felix Pretis, Univ. of Oxford, and James Reade, Univ. of Reading
- GETS modelling of an indicator saturated mean specification
- Indicators: Impulses, steps

Specification:

\[ y_t = \phi_0 + \sum_r \phi_r y_{t-r} + \sum_s \eta_s x_{s,t}^m + \text{indicators} + \epsilon_t, \]

Example of isat w/IIS:

```r
data(Nile)
isat(Nile, ar=1:2, t.pval=0.01)
```

Example of isat w/SIS:

```r
isat(Nile, ar=1:2, sis=TRUE, iis=FALSE, t.pval=0.01)
```

Extraction functions (same as those of getsm):

`coef, fitted, paths, plot, print, residuals, summary, terminals, vcov`
Future versions
Future versions:

- Vignette (a draft available as http://www.sucarrat.net/R/gets.pdf)
- Further speed improvements
- More variance-covariance matrix options
- More flexible block specification and search options in isat
- Multiple rounds?
- From multi-path to multi-branch?
- GETS density modelling?
- New bugs (surprise!)
Thanks!
References:


