

# Package ‘gmm’

January 10, 2010

**Version** 1.3-0

**Date** 2010-01-08

**Title** Generalized Method of Moments and Generalized Empirical Likelihood

**Author** Pierre Chausse <pierre.chausse@uqam.ca>

**Maintainer** Pierre Chausse <pierre.chausse@uqam.ca>

**Description** It is a complete suite to estimate models based on moment conditions. It includes the two step Generalized method of moments (GMM) of Hansen(1982), the iterated GMM and continuous updated estimator (CUE) of Hansen-Eaton-Yaron(1996) and several methods that belong to the Generalized Empirical Likelihood (GEL) family of estimators, as presented by Smith(1997), Kitamura(1997), Newey-Smith(2004) and Anatolyev(2005).

**Depends** R (>= 2.0.0)

**Suggests** mvtnorm, car, fBasics, MASS, timeDate, timeSeries

**Imports** stats

**License** GPL (>= 2)

**Repository** CRAN

**Repository/R-Forge/Project** gmm

**Repository/R-Forge/Revision** 17

**Date/Publication** 2010-01-10 08:04:35

## R topics documented:

charStable . . . . .	2
coef . . . . .	3
confint . . . . .	4
Finance . . . . .	5
FinRes . . . . .	7
fitted . . . . .	7

formula . . . . .	9
gel . . . . .	10
getDat . . . . .	14
getLamb . . . . .	15
getModel . . . . .	16
gmm . . . . .	17
HAC . . . . .	21
kweights2 . . . . .	23
momentEstim . . . . .	23
plot . . . . .	24
print . . . . .	26
residuals . . . . .	27
rho . . . . .	28
smoothG . . . . .	29
specTest . . . . .	31
summary . . . . .	32
vcov . . . . .	34
weightsAndrews2 . . . . .	35

<b>Index</b>	<b>37</b>
--------------	-----------

---

charStable	<i>The characteristic function of a stable distribution</i>
------------	---

---

## Description

It computes the theoretical characteristic function of a stable distribution for two different parametrizations. It is used in the vignette to illustrate the estimation of the parameters using GMM.

## Usage

```
charStable(theta, tau, pm = 0)
```

## Arguments

theta	Vector of parameters of the stable distribution. See details.
tau	A vector of numbers at which the function is evaluated.
pm	The type of parametrization. It takes the values 0 or 1.

## Details

The function returns the vector  $\Psi(\theta, \tau, pm)$  defined as  $E(e^{ix\tau})$ , where  $\tau$  is a vector of real numbers,  $i$  is the imaginary number,  $x$  is a stable random variable with parameters  $\theta = (\alpha, \beta, \gamma, \delta)$  and  $pm$  is the type of parametrization. The vector of parameters are the characteristic exponent, the skewness, the scale and the location parameters, respectively. The restrictions on the parameters are:  $\alpha \in (0, 2]$ ,  $\beta \in [-1, 1]$  and  $\gamma > 0$ . For mode details see Nolan(2009).

**Value**

It returns a vector of complex numbers with the dimension equals to `length(tau)`.

**References**

Nolan J. P. (2009), Stable Distributions. *Math/Stat Department, American University*. URL <http://academic2.american.edu/~jpnolan/stable/stable.html>.

**Examples**

```
# GMM is like GLS for linear models without endogeneity problems

pm <- 0
theta <- c(1.5, .5, 1, 0)
tau <- seq(-3, 3, length.out = 20)
char_fct <- charStable(theta, tau, pm)
```

---

coef

*Coefficients of GEL or GMM*


---

**Description**

It extracts the coefficients from `gel` or `gmm` objects.

**Usage**

```
## S3 method for class 'gmm':
coef(object, ...)
## S3 method for class 'gel':
coef(object, lambda = FALSE, ...)
```

**Arguments**

<code>object</code>	An object of class <code>gel</code> or <code>gmm</code> returned by the function <code>gel</code> or <code>gmm</code>
<code>lambda</code>	If set to <code>TRUE</code> , the lagrange multipliers are extracted instead of the vector of coefficients
<code>...</code>	Other arguments when <code>coef</code> is applied to an other class object

**Value**

Vector of coefficients

## Examples

```
#####
n = 500
phi<-c(.2,.7)
thet <- 0
sd <- .2
x <- matrix(arima.sim(n=n,list(order=c(2,0,1),ar=phi,ma=thet,sd=sd)),ncol=1)
y <- x[7:n]
ym1 <- x[6:(n-1)]
ym2 <- x[5:(n-2)]

H <- cbind(x[4:(n-3)], x[3:(n-4)], x[2:(n-5)], x[1:(n-6)])
g <- y ~ ym1 + ym2
x <- H
t0 <- c(0,.5,.5)

res <- gel(g, x, t0)

coef(res)
coef(res, lambda = TRUE)
#####
res <- gmm(g, x)
coef(res)
```

---

confint

*Confidence intervals for GMM or GEL*

---

## Description

It produces confidence intervals for the coefficients from `gel` or `gmm` estimation.

## Usage

```
## S3 method for class 'gel':
confint(object, parm, level = 0.95, lambda = FALSE, ...)
## S3 method for class 'gmm':
confint(object, parm, level = 0.95, ...)
```

## Arguments

<code>object</code>	An object of class <code>gel</code> or <code>gmm</code> returned by the function <code>gel</code> or <code>gmm</code>
<code>parm</code>	A specification of which parameters are to be given confidence intervals, either a vector of numbers or a vector of names. If missing, all parameters are considered.
<code>level</code>	The confidence level

lambda        If set to TRUE, the confidence intervals for the Lagrange multipliers are produced.

...            Other arguments when confint is applied to another classe object

### Value

It returns a matrix with the first column being the lower bound and the second the upper bound.

### References

Hansen, L.P. (1982), Large Sample Properties of Generalized Method of Moments Estimators. *Econometrica*, **50**, 1029-1054,

Hansen, L.P. and Heaton, J. and Yaron, A.(1996), Finit-Sample Properties of Some Alternative GMM Estimators. *Journal of Business and Economic Statistics*, **14** 262-280.

### Examples

```
#####
n = 500
phi<-c(.2, .7)
thet <- 0
sd <- .2
x <- matrix(arima.sim(n = n, list(order = c(2,0,1), ar = phi, ma = thet, sd = sd)), ncol = 1)
y <- x[7:n]
ym1 <- x[6:(n-1)]
ym2 <- x[5:(n-2)]

H <- cbind(x[4:(n-3)], x[3:(n-4)], x[2:(n-5)], x[1:(n-6)])
g <- y ~ ym1 + ym2
x <- H
t0 <- c(0, .5, .5)

res <- gel(g, x, t0)

confint(res)
confint(res, level = 0.90)
confint(res, lambda = TRUE)

#####

res <- gmm(g, x)

confint(res)
confint(res, level = 0.90)
```

Finance

*Returns on selected stocks***Description**

Daily returns on selected stocks, the Market portfolio and factors of Fama and French from 1993-01-05 to 2009-01-30 for CAPM and APT analysis

**Usage**

```
data(Finance)
```

**Format**

A data frame containing 24 time series. Dates are reported as rownames(). In the following description, company symbols are used.

**WMK** Returns of WEIS MARKETS INC

**UIS** Returns of UNISYS CP NEW

**ORB** Returns of ORBITAL SCIENCES CP

**MAT** Returns of Mattel, Inc.

**ABAX** Returns of ABAXIS, Inc.

**T** Returns of AT&T INC.

**EMR** Returns of EMERSON ELEC CO

**JCS** Returns of Communications Systems Inc.

**VOXX** Returns of Audiovox Corp.

**ZOOM** Returns of ZOOM Technologies Inc.

**TDW** Returns of TIDEWATER INC

**ROG** Returns of Rogers Corporation

**GCG** Returns of Graco Inc.

**PC** Returns of Panasonic Corporation

**GCO** Returns of Genesco Inc.

**EBF** Returns of ENNIS, INC

**F** Returns of FORD MOTOR CO

**FNM** Returns of FANNIE MAE

**NHP** Returns of NATIONWIDE HLTH PROP

**AA** Returns of ALCOA INC

**rf** Risk-free rate of Fama-French

**rm** Return of the market portfolio of Fama-French

**hml** Factor High-Minus-Low of Fama-French

**smb** Factor Small-Minus-Big of Fama-French

**Source**

<http://ca.finance.yahoo.com/> and <http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/>

FinRes

*Method to finalize the result of the momentEstim method***Description**

It computes the final results that will be needed to create the object of class `gmm`).

**Usage**

```
## S3 method for class 'baseGmm.res':
FinRes(z, object, ...)
```

**Arguments**

<code>z</code>	An object of class determined by the method <code>momentEstim</code> .
<code>object</code>	An object produced by <code>getModel</code>
<code>...</code>	Other argument to be passed to other <code>FinRes</code> methods.

**Value**

It returns an object of class `gmm`. See [gmm](#) for more details.

**References**

Hansen, L.P. (1982), Large Sample Properties of Generalized Method of Moments Estimators. *Econometrica*, **50**, 1029-1054,  
 Hansen, L.P. and Heaton, J. and Yaron, A.(1996), Finit-Sample Properties of Some Alternative GMM Estimators. *Journal of Business and Economic Statistics*, **14** 262-280.

fitted

*Fitted values of GEL and GMM***Description**

Method to extract the fitted values of the model estimated by [gel](#) or [gmm](#).

**Usage**

```
## S3 method for class 'gel':
fitted(object, ...)
## S3 method for class 'gmm':
fitted(object, ...)
```

**Arguments**

object            An object of class `gel` or `gel` returned by the function `gel` or `gmm`  
 ...                Other arguments when `fitted` is applied to an other class object

**Value**

It returns a matrix of the estimated mean  $\hat{y}$  in  $g=y\sim x$  as it is done by `fitted.lm`.

**Examples**

```
# GEL can deal with endogeneity problems

n = 200
phi<-c(.2,.7)
thet <- 0.2
sd <- .2
set.seed(123)
x <- matrix(arima.sim(n = n, list(order = c(2,0,1), ar = phi, ma = thet, sd = sd)), ncol = 1)

y <- x[7:n]
ym1 <- x[6:(n-1)]
ym2 <- x[5:(n-2)]
H <- cbind(x[4:(n-3)], x[3:(n-4)], x[2:(n-5)], x[1:(n-6)])
g <- y ~ ym1 + ym2
x <- H

res <- gel(g, x, c(0,.3,.6))
plot(y, main = "Fitted ARMA with GEL")
lines(fitted(res), col = 2)

# GMM is like GLS for linear models without endogeneity problems

set.seed(345)
n = 200
phi<-c(.2,.7)
thet <- 0
sd <- .2
x <- matrix(arima.sim(n = n, list(order = c(2,0,1), ar = phi, ma = thet, sd = sd)), ncol = 1)
y <- 10 + 5*rnorm(n) + x

res <- gmm(y ~ x, x)
plot(x, y, main = "Fitted model with GMM")
lines(x, fitted(res), col = 2)
legend("topright", c("Y", "Yhat"), col = 1:2, lty = c(1,1))
```

---

 formula

*Formula method for gel and gmm objects*


---

**Description**

Method to extract the formula from `gel` or `gmm` objects.

**Usage**

```
## S3 method for class 'gel':
formula(x, ...)
## S3 method for class 'gmm':
formula(x, ...)
```

**Arguments**

`x`                    An object of class `gel` or `gmm` returned by the function `gel` or `gmm`  
`...`                 Other arguments to pass to other methods

**Examples**

```
## GEL ##
n = 200
phi<-c(.2,.7)
thet <- 0.2
sd <- .2
set.seed(123)
x <- matrix(arima.sim(n = n, list(order = c(2,0,1), ar = phi, ma = thet, sd = sd)), ncol = 1)

y <- x[7:n]
ym1 <- x[6:(n-1)]
ym2 <- x[5:(n-2)]
H <- cbind(x[4:(n-3)], x[3:(n-4)], x[2:(n-5)], x[1:(n-6)])
g <- y ~ ym1 + ym2
x <- H

res <- gel(g, x, c(0,.3,.6))
formula(res)

# GMM is like GLS for linear models without endogeneity problems

set.seed(345)
n = 200
phi<-c(.2,.7)
thet <- 0
sd <- .2
x <- matrix(arima.sim(n = n, list(order = c(2,0,1), ar = phi, ma = thet, sd = sd)), ncol = 1)
y <- 10 + 5*rnorm(n) + x
```

```
res <- gmm(y ~ x, x)
formula(res)
```

gel

*Generalized Empirical Likelihood estimation***Description**

Function to estimate a vector of parameters based on moment conditions using the GEL method as presented by Newey-Smith(2004) and Anatolyev(2005).

**Usage**

```
gel(g, x, tet0, gradv = NULL, smooth = FALSE, type = c("EL", "ET", "CUE", "ETEL"),
    kernel = c("Truncated", "Bartlett"), bw = bwAndrews2,
    approx = c("AR(1)", "ARMA(1,1)"), prewhite = 1, ar.method = "ols", tol_weights
    tol_lam = 1e-9, tol_obj = 1e-9, tol_mom = 1e-9, maxiterlam = 100, constraint =
    optfct = c("optim", "optimize", "nlminb"), optlam = c("iter", "numeric"), model
    X = FALSE, Y = FALSE, TypeGel = "baseGel", ...)
```

**Arguments**

g	A function of the form $g(\theta, x)$ and which returns a $n \times q$ matrix with typical element $g_i(\theta, x_t)$ for $i = 1, \dots, q$ and $t = 1, \dots, n$ . This matrix is then used to build the $q$ sample moment conditions. It can also be a formula if the model is linear (see details below).
tet0	A $k \times 1$ vector of starting values. If the dimension of $\theta$ is one, see the argument "optfct".
x	The matrix or vector of data from which the function $g(\theta, x)$ is computed. If "g" is a formula, it is an $n \times Nh$ matrix of instruments (see details below).
gradv	A function of the form $G(\theta, x)$ which returns a $q \times k$ matrix of derivatives of $\bar{g}(\theta)$ with respect to $\theta$ . By default, the numerical algorithm <code>numericDeriv</code> is used. It is of course strongly suggested to provide this function when it is possible. This gradient is used compute the asymptotic covariance matrix of $\hat{\theta}$ . If "g" is a formula, the gradient is not required (see the details below).
smooth	If set to TRUE, the moment function is smoothed as proposed by Kitamura(1997)
type	"EL" for empirical likelihood, "ET" for exponential tilting, "CUE" for continuous updated estimator and "ETEL" for exponentially tilted empirical likelihood of Schennach(2007).
kernel	type of kernel used to compute the covariance matrix of the vector of sample moment conditions (see <a href="#">HAC</a> for more details) and to smooth the moment conditions if "smooth" is set to TRUE. Only two types of kernel are available. The truncated implies a Bartlett kernel for the HAC matrix and the Bartlett implies a Parzen kernel (see Smith 2004).

<code>bw</code>	The method to compute the bandwidth parameter. By default it is <code>bwAndrews2</code> which is proposed by Andrews (1991). The alternative is <code>bwNeweyWest2</code> of Newey-West(1994).
<code>prewhite</code>	logical or integer. Should the estimating functions be prewhitened? If TRUE or greater than 0 a VAR model of order <code>as.integer(prewhite)</code> is fitted via <code>ar</code> with method "ols" and <code>demean = FALSE</code> .
<code>ar.method</code>	character. The <code>method</code> argument passed to <code>ar</code> for prewhitening.
<code>approx</code>	a character specifying the approximation method if the bandwidth has to be chosen by <code>bwAndrews2</code> .
<code>tol_weights</code>	numeric. Weights that exceed <code>tol</code> are used for computing the covariance matrix, all other weights are treated as 0.
<code>tol_lam</code>	Tolerance for $\lambda$ between two iterations. The algorithm stops when $\ \lambda_i - \lambda_{i-1}\ $ reaches <code>tol_lamb</code> (see <code>getLamb</code> )
<code>maxiterlam</code>	The algorithm to compute $\lambda$ stops if there is no convergence after "maxiterlam" iterations (see <code>getLamb</code> ).
<code>tol_obj</code>	Tolerance for the gradient of the objective function to compute $\lambda$ (see <code>getLamb</code> ).
<code>optfct</code>	Only when the dimension of $\theta$ is 1, you can choose between the algorithm <code>optim</code> or <code>optimize</code> . In that case, the former is unreliable. If <code>optimize</code> is chosen, "t0" must be $1 \times 2$ which represents the interval in which the algorithm seeks the solution. It is also possible to choose the <code>nlminb</code> algorithm. In that case, bounds for the coefficients can be set by the options <code>upper=</code> and <code>lower=</code> .
<code>constraint</code>	If set to TRUE, the constraint optimization algorithm is used. See <code>constrOptim</code> to learn how it works. In particular, if you choose to use it, you need to provide "ui" and "ci" in order to impose the constraint $ui\theta - ci \geq 0$ .
<code>tol_mom</code>	It is the tolerance for the moment condition $\sum_{t=1}^n p_t g(\theta(x_t)) = 0$ , where $p_t = \frac{1}{n} D\rho(< g_t, \lambda >)$ is the implied probability. It adds a penalty if the solution diverges from its goal.
<code>optlam</code>	The default is "iter" which solves for $\lambda$ using the Newton iterative method <code>getLamb</code> . If set to "numeric", the algorithm <code>optim</code> is used to compute $\lambda$ instead.
<code>model, X, Y</code>	logicals. If TRUE the corresponding components of the fit (the model frame, the model matrix, the response) are returned if <code>g</code> is a formula.
<code>TypeGel</code>	The name of the class object created by the method <code>getModel</code> . It allows developers to extend the package and create other GEL methods.
<code>...</code>	More options to give to <code>optim</code> , <code>optimize</code> or <code>constrOptim</code> .

## Details

`weightsAndrews2` and `bwAndrews2` are simply modified version of `weightsAndrews` and `bwAndrews` from the package `sandwich`. The modifications have been made so that the argument `x` can be a matrix instead of an object of class `lm` or `glm`. The details on how it works can be found on the `sandwich` manual.

If we want to estimate a model like  $Y_t = \theta_1 + X_{2t}\theta_2 + \dots + X_k\theta_k + \epsilon_t$  using the moment conditions  $Cov(\epsilon_t H_t) = 0$ , where  $H_t$  is a vector of  $Nh$  instruments, than we can define "g" like we do for `lm`.

We would have  $g = y - x_2 + x_3 + \dots + x_k$  and the argument "x" above would become the matrix  $H$  of instruments. As for `lm`,  $Y_t$  can be a  $Ny \times 1$  vector which would imply that  $k = Nh \times Ny$ . The intercept is included by default so you do not have to add a column of ones to the matrix  $H$ . You do not need to provide the gradient in that case since in that case it is embedded in `gel`. The intercept can be removed by adding -1 to the formula. In that case, the column of ones need to be added manually to  $H$ .

If "smooth" is set to TRUE, the sample moment conditions  $\sum_{t=1}^n g(\theta, x_t)$  is replaced by:  $\sum_{t=1}^n g^k(\theta, x_t)$ , where  $g^k(\theta, x_t) = \sum_{i=-r}^r k(i)g(\theta, x_{t+i})$ , where  $r$  is a truncated parameter that depends on the bandwidth and  $k(i)$  are normalized weights so that they sum to 1.

The method solves  $\hat{\theta} = \arg \min [\arg \max_{\lambda} \frac{1}{n} \sum_{t=1}^n \rho(< g(\theta, x_t), \lambda >) - \rho(0)]$

## Value

'gel' returns an object of 'class' "'gel'"

The functions 'summary' is used to obtain and print a summary of the results.

The object of class "gel" is a list containing at least the following:

<code>coefficients</code>	$k \times 1$ vector of parameters
<code>residuals</code>	the residuals, that is response minus fitted values if "g" is a formula.
<code>fitted.values</code>	the fitted mean values if "g" is a formula.
<code>lambda</code>	$q \times 1$ vector of Lagrange multipliers.
<code>vcov_par</code>	the covariance matrix of "coefficients"
<code>vcov_lambda</code>	the covariance matrix of "lambda"
<code>pt</code>	The implied probabilities
<code>objective</code>	the value of the objective function
<code>conv_lambda</code>	Convergence code for "lambda" (see <code>getLamb</code> )
<code>conv_mes</code>	Convergence message for "lambda" (see <code>getLamb</code> )
<code>conv_par</code>	Convergence code for "coefficients" (see <code>optim</code> , <code>optimize</code> or <code>constrOptim</code> )
<code>terms</code>	the <code>terms</code> object used when g is a formula.
<code>call</code>	the matched call.
<code>y</code>	if requested, the response used (if "g" is a formula).
<code>x</code>	if requested, the model matrix used if "g" is a formula or the data if "g" is a function.
<code>model</code>	if requested (the default), the model frame used if "g" is a formula.

## References

Anatolyev, S. (2005), GMM, GEL, Serial Correlation, and Asymptotic Bias. *Econometrica*, **73**, 983-1002.

Andrews DWK (1991), Heteroskedasticity and Autocorrelation Consistent Covariance Matrix Estimation. *Econometrica*, **59**, 817-858.

Kitamura, Yuichi (1997), Empirical Likelihood Methods With Weakly Dependent Processes. *The Annals of Statistics*, **25**, 2084-2102.

Newey, W.K. and Smith, R.J. (2004), Higher Order Properties of GMM and Generalized Empirical Likelihood Estimators. *Econometrica*, **72**, 219-255.

Smith, R.J. (2004), GEL Criteria for Moment Condition Models. *Working paper, CEMMAP*.

Newey WK & West KD (1987), A Simple, Positive Semi-Definite, Heteroskedasticity and Autocorrelation Consistent Covariance Matrix. *Econometrica*, **55**, 703-708.

Newey WK & West KD (1994), Automatic Lag Selection in Covariance Matrix Estimation. *Review of Economic Studies*, **61**, 631-653.

Schennach, Susanne, M. (2007), Point Estimation with Exponentially Tilted Empirical Likelihood. *Econometrica*, **35**, 634-672.

Zeileis A (2006), Object-oriented Computation of Sandwich Estimators. *Journal of Statistical Software*, **16**(9), 1-16. URL <http://www.jstatsoft.org/v16/i09/>.

## Examples

```
# First, an exemple with the fonction g()

g <- function(tet, x)
{
  n <- nrow(x)
  u <- (x[7:n] - tet[1] - tet[2]*x[6:(n-1)] - tet[3]*x[5:(n-2)])
  f <- cbind(u, u*x[4:(n-3)], u*x[3:(n-4)], u*x[2:(n-5)], u*x[1:(n-6)])
  return(f)
}

Dg <- function(tet,x)
{
  n <- nrow(x)
  xx <- cbind(rep(1, (n-6)), x[6:(n-1)], x[5:(n-2)])
  H <- cbind(rep(1, (n-6)), x[4:(n-3)], x[3:(n-4)], x[2:(n-5)], x[1:(n-6)])
  f <- -crossprod(H, xx)/(n-6)
  return(f)
}

n = 200
phi<-c(.2, .7)
thet <- 0.2
sd <- .2
set.seed(123)
x <- matrix(arima.sim(n = n, list(order = c(2, 0, 1), ar = phi, ma = thet, sd = sd)), ncol =

res <- gel(g, x, c(0, .3, .6), grad = Dg)
summary(res)

# The same model but with g as a formula.... much simpler in that case

y <- x[7:n]
ym1 <- x[6:(n-1)]
ym2 <- x[5:(n-2)]
```

```
H <- cbind(x[4:(n-3)], x[3:(n-4)], x[2:(n-5)], x[1:(n-6)])
g <- y ~ ym1 + ym2
x <- H

res <- gel(g, x, c(0, .3, .6))
summary(res)
```

---

getDat

*Extracting data from a formula*


---

## Description

It extract the data from a formula  $y \sim z$  with instrument  $h$  and put everything in a matrix. It helps redefine the function  $g(\theta, x)$  that is required by `gmm` and `gel`.

## Usage

```
getDat(formula, h)
```

## Arguments

`formula`      A formula that defines the linear model to be estimated (see details).  
`h`              A  $n \times nh$  matrix of intruments(see details).

## Details

The model to be estimated is based on the moment conditions  $\langle h, (y - z\theta) \rangle = 0$ . It adds a column of ones to  $z$  and  $h$  by default. They are removed if -1 is added to the formula

## Value

`x`: A  $n \times l$  matrix, where  $l = ncol(y) + ncol(z) + ncol(h) + 2$  if "intercept" is TRUE and  $ncol(y) + ncol(z) + ncol(h)$  if "intercept" is FALSE.

`nh`: dimension of  $h$

`k`: dimension of  $z$

`ny`: dimension of  $y$

## Examples

```
n = 500
phi <- c(.2, .7)
thet <- 0.2
sd <- .2
x <- matrix(arima.sim(n = n, list(order = c(2, 0, 1), ar = phi, ma = thet, sd = sd)), ncol =
y <- x[7:n]
ym1 <- x[6:(n-1)]
ym2 <- x[5:(n-2)]
```

```
H <- cbind(x[4:(n-3)], x[3:(n-4)], x[2:(n-5)], x[1:(n-6)])
x <- getDat(y ~ ym1 + ym2, H)
```

---

getLamb *Solving for the Lagrange multipliers of Generalized Empirical Likelihood (GEL)*

---

### Description

It computes the vector of Lagrange multipliers, which maximizes the GEL objective function, using an iterative Newton method.

### Usage

```
getLamb(g, tet, x, type = c('EL', 'ET', 'CUE'), tol_lam = 1e-12, maxiterlam = 1000, t
```

### Arguments

g	A function of the form $g(\theta, x)$ and which returns a $n \times q$ matrix with typical element $g_i(\theta, x_t)$ for $i = 1, \dots, q$ and $t = 1, \dots, n$ . This matrix is then used to build the $q$ sample moment conditions.
tet	A $k \times 1$ vector of parameters at which the function $g(\theta, x)$ has to be evaluated
x	The matrix or vector of data from which the function $g(\theta, x)$ is computed.
type	"EL" for empirical likelihood, "ET" for exponential tilting and "CUE" for continuous updated estimator.
tol_lam	Tolerance for $\lambda$ between two iterations. The algorithm stops when $\ \lambda_i - \lambda_{i-1}\ $ reaches <code>tol_lam</code>
maxiterlam	The algorithm stops if there is no convergence after "maxiterlam" iterations.
tol_obj	Tolerance for the gradient of the objective function. The algorithm returns a non-convergence message if $\max( gradient )$ does not reach <code>tol_obj</code> . It helps the <code>gel</code> algorithm to select the right space to look for $\theta$
k	It represents the ratio $k_1/k_2$ , where $k_1 = \int_{-\infty}^{\infty} k(s)ds$ and $k_2 = \int_{-\infty}^{\infty} k(s)^2 ds$ . See Smith(2004).

### Details

It solves the problem  $\frac{1}{n} \sum_{t=1}^n D\rho(\langle g(\theta, x_t), \lambda \rangle) g(\theta, x_t) = 0$ .

### Value

lambda: A  $q \times 1$  vector of Lagrange multipliers which solve the system of equations given above.  
singular: 0 for a normal solution, 1 if the algorithm does not converge and 2 if the algorithm produces a singular system, NaN or Inf values.  
conv\_mes: A message with details about the convergence.

## References

- Newey, W.K. and Smith, R.J. (2004), Higher Order Properties of GMM and Generalized Empirical Likelihood Estimators. *Econometrica*, **72**, 219-255.
- Smith, R.J. (2004), GEL Criteria for Moment Condition Models. *Working paper, CEMMAP*.

## Examples

```
g <- function(tet, x)
{
  n <- nrow(x)
  u <- (x[7:n] - tet[1] - tet[2]*x[6:(n-1)] - tet[3]*x[5:(n-2)])
  f <- cbind(u, u*x[4:(n-3)], u*x[3:(n-4)], u*x[2:(n-5)], u*x[1:(n-6)])
  return(f)
}
n = 500
phi<-c(.2, .7)
thet <- 0.2
sd <- .2
x <- matrix(arima.sim(n = n, list(order = c(2, 0, 1), ar = phi, ma = thet, sd = sd)), ncol =
getLamb(g, c(0, phi), x, type = "EL")
```

---

getModel

*Method for setting the properties of a model*

---

## Description

It collects what is needed by the method `momentEstim` (see details).

## Usage

```
## S3 method for class 'baseGmm':
getModel(object, ...)
```

## Arguments

<code>object</code>	An object of class <code>baseGmm</code>
<code>...</code>	Other arguments when <code>getModel</code> is applied to another class object

## Value

It returns an object of the right class which determines how the method `momentEstim` will treat it. For example, if `g` is a formula and `type` is set to "cue", it creates an object of class `baseGmm.cue.formula`. In this case, `momentEstim`, applied to this object, computes the continuously updated GMM of a linear model. It allows more flexibility this way. For example, it could be easy to add a GMM method which is robust in presence of weak identification simply by creating a new class of model and the associated `momentEstim` method.

## Description

Function to estimate a vector of parameters based on moment conditions using the GMM method of Hansen(82).

## Usage

```
gmm(g, x, t0 = NULL, gradv = NULL, type = c("twoStep", "cue", "iterative"),
    wmatrix = c("optimal", "ident"), vcov = c("HAC", "iid"),
    kernel = c("Quadratic Spectral", "Truncated", "Bartlett",
              "Parzen", "Tukey-Hanning"), crit = 10e-7, bw = bwAndrews2,
    prewhite = FALSE, ar.method = "ols", approx = "AR(1)", tol = 1e-7,
    itermax = 100, optfct = c("optim", "optimize", "nlnminb"), model=TRUE, X=FALSE, Y=FALSE)
```

## Arguments

<code>g</code>	A function of the form $g(\theta, x)$ and which returns a $n \times q$ matrix with typical element $g_i(\theta, x_t)$ for $i = 1, \dots, q$ and $t = 1, \dots, n$ . This matrix is then used to build the $q$ sample moment conditions. It can also be a formula if the model is linear (see details below).
<code>x</code>	The matrix or vector of data from which the function $g(\theta, x)$ is computed. If "g" is a formula, it is an $n \times Nh$ matrix of instruments (see details below).
<code>t0</code>	A $k \times 1$ vector of starting values. It is required only when "g" is a function because only then a numerical algorithm is used to minimize the objective function. If the dimension of $\theta$ is one, see the argument "optfct".
<code>gradv</code>	A function of the form $G(\theta, x)$ which returns a $q \times k$ matrix of derivatives of $\bar{g}(\theta)$ with respect to $\theta$ . By default, the numerical algorithm <code>numericDeriv</code> is used. It is of course strongly suggested to provide this function when it is possible. This gradient is used compute the asymptotic covariance matrix of $\hat{\theta}$ . If "g" is a formula, the gradient is not required (see the details below).
<code>type</code>	The GMM method: "twoStep" is the two step GMM proposed by Hansen(1982) and the "cue" and "iterative" are respectively the continuous updated and the iterative GMM proposed by Hansen, Eaton et Yaron (1996)
<code>wmatrix</code>	Which weighting matrix should be used in the objective function. By default, it is the inverse of the covariance matrix of $g(\theta, x)$ . The other choice is the identity matrix which is usually used to obtain a first step estimate of $\theta$
<code>vcov</code>	Assumption on the properties of the random vector $x$ . By default, $x$ is a weakly dependant process. The "iid" option will avoid using the HAC matrix which will accelerate the estimation if one is ready to make that assumption.
<code>kernel</code>	type of kernel used to compute the covariance matrix of the vector of sample moment conditions (see <a href="#">HAC</a> for more details)

<code>crit</code>	The stopping rule for the iterative GMM. It can be reduce to increase the precision.
<code>bw</code>	The method to compute the bandwidth parameter. By default it is <code>bwAndrews2</code> which is proposed by Andrews (1991). The alternative is <code>bwNeweyWest2</code> of Newey-West(1994).
<code>prewhite</code>	logical or integer. Should the estimating functions be prewhitened? If TRUE or greater than 0 a VAR model of order <code>as.integer(prewhite)</code> is fitted via <code>ar</code> with method "ols" and <code>demean = FALSE</code> .
<code>ar.method</code>	character. The <code>method</code> argument passed to <code>ar</code> for prewhitening.
<code>approx</code>	A character specifying the approximation method if the bandwidth has to be chosen by <code>bwAndrews2</code> .
<code>tol</code>	Weights that exceed <code>tol</code> are used for computing the covariance matrix, all other weights are treated as 0.
<code>itermax</code>	The maximum number of iterations for the iterative GMM. It is unlikely that the algorithm does not converge but we keep it as a safety.
<code>optfct</code>	Only when the dimension of $\theta$ is 1, you can choose between the algorithm <code>optim</code> or <code>optimize</code> . In that case, the former is unreliable. If <code>optimize</code> is chosen, "t0" must be $1 \times 2$ which represents the interval in which the algorithm seeks the solution. It is also possible to choose the <code>nlminb</code> algorithm. In that case, borns for the coefficients can be set by the options <code>upper=</code> and <code>lower=</code> .
<code>model, X, Y</code>	logicals. If TRUE the corresponding components of the fit (the model frame, the model matrix, the response) are returned if <code>g</code> is a formula.
<code>TypeGmm</code>	The name of the class object created by the method <code>getModel</code> . It allows developers to extend the package and create other GMM methods.
<code>...</code>	More options to give to <code>optim</code> .

## Details

`weightsAndrews2` and `bwAndrews2` are simply modified version of `weightsAndrews` and `bwAndrews` from the package `sandwich`. The modifications have been made so that the argument `x` can be a matrix instead of an object of class `lm` or `glm`. The details on how is works can be found on the `sandwich` manual.

If we want to estimate a model like  $Y_t = \theta_1 + X_{2t}\theta_2 + \dots + X_k\theta_k + \epsilon_t$  using the moment conditions  $Cov(\epsilon_t H_t) = 0$ , where  $H_t$  is a vector of  $Nh$  instruments, than we can define "g" like we do for `lm`. We would have  $g = y \sim x_2 + x_3 + \dots + x_k$  and the argument "x" above would become the matrix  $H$  of instruments. As for `lm`,  $Y_t$  can be a  $Ny \times 1$  vector which would imply that  $k = Nh \times Ny$ . The intercept is included by default so you do not have to add a column of ones to the matrix  $H$ . You do not need to provide the gradient in that case since in that case it is embedded in `gmm`. The intercept can be removed by adding -1 to the formula. In that case, the column of ones need to be added manually to  $H$ .

The following explains the last example bellow. Thanks to Dieter Rozenich, a student from the Vienna Universtiy of Economics and Business Administration. He suggested that it would help to understand the implementation of the jacobian.

For the two parameters of a normal distribution  $(\mu, \sigma)$  we have the following three moment conditions:

$$\begin{aligned}m_1 &= \mu - x_i \\m_2 &= \sigma^2 - (x_i - \mu)^2 \\m_3 &= x_i^3 - \mu(\mu^2 + 3\sigma^2)\end{aligned}$$

$m_1, m_2$  can be directly obtained by the definition of  $(\mu, \sigma)$ . The third moment condition comes from the third derivative of the moment generating function (MGF)

$$M_X(t) = \exp\left(\mu t + \frac{\sigma^2 t^2}{2}\right)$$

evaluated at  $(t = 0)$ .

Note that we have more equations (3) than unknown parameters (2).

The Jacobian of these two conditions is (it should be an array but I can't make it work):

$$\begin{array}{cc}1 & 0 \\-2\mu + 2x & 2\sigma \\-3\mu^2 - 3\sigma^2 & -6\mu\sigma\end{array}$$

## Value

'gmm' returns an object of 'class' "gmm"

The functions 'summary' is used to obtain and print a summary of the results. It also compute the J-test of overidentying restriction

The object of class "gmm" is a list containing at least:

coefficients	$k \times 1$ vector of coefficients
residuals	the residuals, that is response minus fitted values if "g" is a formula.
fitted.values	the fitted mean values if "g" is a formula.
vcov	the covariance matrix of the coefficients
objective	the value of the objective function $\ var(\bar{g})^{-1/2}\bar{g}\ ^2$
terms	the <code>terms</code> object used when g is a formula.
call	the matched call.
y	if requested, the response used (if "g" is a formula).
x	if requested, the model matrix used if "g" is a formula or the data if "g" is a function.
model	if requested (the default), the model frame used if "g" is a formula.

## References

Zeileis A (2006), Object-oriented Computation of Sandwich Estimators. *Journal of Statistical Software*, **16**(9), 1–16. URL <http://www.jstatsoft.org/v16/i09/>.

Andrews DWK (1991), Heteroskedasticity and Autocorrelation Consistent Covariance Matrix Estimation. *Econometrica*, **59**, 817–858.

Newey WK & West KD (1987), A Simple, Positive Semi-Definite, Heteroskedasticity and Autocorrelation Consistent Covariance Matrix. *Econometrica*, **55**, 703–708.

Newey WK & West KD (1994), Automatic Lag Selection in Covariance Matrix Estimation. *Review of Economic Studies*, **61**, 631-653.

Hansen, L.P. (1982), Large Sample Properties of Generalized Method of Moments Estimators. *Econometrica*, **50**, 1029-1054,

Hansen, L.P. and Heaton, J. and Yaron, A.(1996), Finit-Sample Properties of Some Alternative GMM Estimators. *Journal of Business and Economic Statistics*, **14** 262-280.

## Examples

```
## CAPM test with GMM
data(Finance)
r <- Finance[1:300, 1:10]
rm <- Finance[1:300, "rm"]
rf <- Finance[1:300, "rf"]

z <- as.matrix(r-rf)
t <- nrow(z)
zm <- rm-rf
h <- matrix(zm, t, 1)
res <- gmm(z ~ zm, x = h)
summary(res)

## linear tests can be performed using linear.hypothesis from the car package
## The CAPM can be tested as follows:

library(car)
linear.hypothesis(res,cbind(diag(10),matrix(0,10,10)),rep(0,10))

# The CAPM of Black
g <- function(theta, x) {
e <- x[,2:11] - theta[1] - (x[,1] - theta[1]) %*% matrix(theta[2:11], 1, 10)
gmat <- cbind(e, e*c(x[,1]))
return(gmat) }

x <- as.matrix(cbind(rm, r))
res_black <- gmm(g, x = x, t0 = rep(0, 11))

summary(res_black)$coefficients

## APT test with Fama-French factors and GMM
```

```

f1 <- zm
f2 <- Finance[1:300, "hml"] - rf
f3 <- Finance[1:300, "smb"] - rf
h <- cbind(f1, f2, f3)
res2 <- gmm(z ~ f1 + f2 + f3, x = h)
coef(res2)
summary(res2)$coefficients

## The following example has been provided by Dieter Rozenich (see details).
# It generates normal random numbers and uses the GMM to estimate
# mean and sd.
#-----
# Random numbers of a normal distribution
# First we generate normally distributed random numbers and compute the two parameters:
n <- 1000
x <- rnorm(n, mean = 4, sd = 2)
# Implementing the 3 moment conditions
g <- function(tet, x)
{
  m1 <- (tet[1] - x)
  m2 <- (tet[2]^2 - (x - tet[1])^2)
  m3 <- x^3 - tet[1]*(tet[1]^2 + 3*tet[2]^2)
  f <- cbind(m1, m2, m3)
  return(f)
}
# Implementing the jacobian
Dg <- function(tet, x)
{
  jacobian <- matrix(c( 1, 2*(-tet[1]+mean(x)), -3*tet[1]^2-3*tet[2]^2, 0, 2*tet[2], -6*x),
  return(jacobian)
}
# Now we want to estimate the two parameters using the GMM.
gmm(g, x, c(0, 0), grad = Dg)

```

---

HAC

*Covariance matrix of weakly dependent time series*


---

### Description

Function to compute a consistent covariance matrix of the sample mean of a random vector of time series. It is consistent in presence of heteroscedasticity and autocorrelation.

### Usage

```

HAC(x, weights = weightsAndrews2, bw = bwAndrews2, prewhite = FALSE, ar.method = "c",
kernel=c("Quadratic Spectral", "Truncated", "Bartlett", "Parzen", "Tukey-Hanning"),
approx = "AR(1)", tol = 1e-7)

```

### Arguments

<code>x</code>	a $n \times q$ matrix of time series, where $n$ is the sample size.
<code>weights</code>	The method to compute the kernel weights. For now, <code>weightsAndrews2</code> is the only one possible. I leave the option there because I am planning to give more choices in futur versions of the package.
<code>bw</code>	The method to compute the bandwidth parameter. By default it is <code>bwAndrews2</code> which is proposed by Andrews (1991). The alternative is <code>bwNeweyWest2</code> of Newey-West(1994).
<code>prewhite</code>	logical or integer. Should the estimating functions be prewhitened? If TRUE or greater than 0 a VAR model of order <code>as.integer(prewhite)</code> is fitted via <code>ar</code> with method "ols" and <code>demean = FALSE</code> .
<code>ar.method</code>	character. The <code>method</code> argument passed to <code>ar</code> for prewhitening.
<code>kernel</code>	The choice of kernel
<code>approx</code>	a character specifying the approximation method if the bandwidth has to be chosen by <code>bwAndrews2</code> .
<code>tol</code>	numeric. Weights that exceed <code>tol</code> are used for computing the covariance matrix, all other weights are treated as 0.

### Details

HAC is simply a modified version of `meatHAC` from the package `sandwich`. The modifications have been made so that the argument `x` can be a matrix instead of an object of class `lm` or `glm`. The details on how is works can be found on the `sandwich` manual.

### Value

A  $q \times q$  matrix containing an estimator of the asymptotic variance of  $\sqrt{n}\bar{x}$ , where  $\bar{x}$  is  $q \times 1$  vector with typical element  $\bar{x}_i = \frac{1}{n} \sum_{j=1}^n x_{ji}$ . This function is called by `gmm()` but can also be used by itself.

### References

- Zeileis A (2006), Object-oriented Computation of Sandwich Estimators. *Journal of Statistical Software*, **16**(9), 1–16. URL <http://www.jstatsoft.org/v16/i09/>.
- Newey WK & West KD (1987), A Simple, Positive Semi-Definite, Heteroskedasticity and Autocorrelation Consistent Covariance Matrix. *Econometrica*, **55**, 703–708.
- Newey WK & West KD (1994), Automatic Lag Selection in Covariance Matrix Estimation. *Review of Economic Studies*, **61**, 631-653.

### Examples

```
x <- arima.sim(n = 200, list(order = c(1, 0, 1), ar = .5, ma = .3, sd = .5))
y <- .4*x + rnorm(200)
x <- cbind(x, y)
vcov <- HAC(x)
```

---

kweights2

*Kernel Weights*


---

### Description

Kernel weights for kernel-based heteroskedasticity and autocorrelation consistent (HAC) covariance matrix estimators as introduced by Andrews (1991). This function comes from the `sandwich` package.

### Usage

```
kweights2(x, kernel = c("Truncated", "Bartlett", "Parzen",
  "Tukey-Hanning", "Quadratic Spectral"), normalize = FALSE)
```

### Arguments

<code>x</code>	numeric.
<code>kernel</code>	a character specifying the kernel used. All kernels used are described in Andrews (1991).
<code>normalize</code>	logical. If set to <code>TRUE</code> the kernels are normalized as described in Andrews (1991).

### Value

Value of the kernel function at `x`.

### References

Andrews DWK (1991), Heteroskedasticity and Autocorrelation Consistent Covariance Matrix Estimation. *Econometrica*, **59**, 817–858.

---

momentEstim

*Method for estimating models based on moment conditions*


---

### Description

It estimates a model which is characterized by the method `getModel` (see details).

**Usage**

```
## S3 method for class 'baseGmm.twoStep':
momentEstim(object, ...)
## S3 method for class 'baseGmm.twoStep.formula':
momentEstim(object, ...)
## S3 method for class 'baseGmm.iterative.formula':
momentEstim(object, ...)
## S3 method for class 'baseGmm.iterative':
momentEstim(object, ...)
## S3 method for class 'baseGmm.cue.formula':
momentEstim(object, ...)
## S3 method for class 'baseGmm.cue':
momentEstim(object, ...)
```

**Arguments**

`object`            An object created by the method `getModel`  
`...`             Other arguments when `momentEstim` is applied to an other class object

**Value**

It returns an object of class determined by the argument "TypeGMM" of `gmm`. By default, it is of class `baseGmm.res`. It estimates the model and organize the results that will be finalized by the method `FinRes`. More methods can be created in order to use other GMM methods not yet included in the package.

**References**

Hansen, L.P. (1982), Large Sample Properties of Generalized Method of Moments Estimators. *Econometrica*, **50**, 1029-1054,  
Hansen, L.P. and Heaton, J. and Yaron, A.(1996), Finit-Sample Properties of Some Alternative GMM Estimators. *Journal of Business and Economic Statistics*, **14** 262-280.

---

plot

*Plot Diagnostics for gel and gmm objects*


---

**Description**

It is a plot method for `gel` or `gmm` objects.

**Usage**

```
## S3 method for class 'gel':
plot(x, which = c(1L:4),
      main = list("Residuals vs Fitted values", "Normal Q-Q",
                 "Response variable and fitted values", "Implied probabilities"),
```

```

panel = if(add.smooth) panel.smooth else points,
ask = prod(par("mfcol")) < length(which) && dev.interactive(), ...,
add.smooth = getOption("add.smooth")

## S3 method for class 'gmm':
plot(x, which = c(1L:3),
     main = list("Residuals vs Fitted values", "Normal Q-Q",
                "Response variable and fitted values"),
     panel = if(add.smooth) panel.smooth else points,
     ask = prod(par("mfcol")) < length(which) && dev.interactive(), ...,
     add.smooth = getOption("add.smooth"))

```

### Arguments

<code>x</code>	gel or gmm object, typically result of <code>gel</code> or <code>gmm</code> .
<code>which</code>	if a subset of the plots is required, specify a subset of the numbers 1:4 for <code>gel</code> or 1:3 for <code>gmm</code> .
<code>main</code>	Vector of titles for each plot.
<code>panel</code>	panel function. The useful alternative to <code>points</code> , <code>panel.smooth</code> can be chosen by <code>add.smooth = TRUE</code> .
<code>ask</code>	logical; if TRUE, the user is <i>asked</i> before each plot, see <code>par(ask=.)</code> .
<code>...</code>	other parameters to be passed through to plotting functions.
<code>add.smooth</code>	logical indicating if a smoother should be added to most plots; see also <code>panel</code> above.

### Details

It is a beta version of a plot method for `gel` objects. It is a modified version of `plot.lm`. For now, it is available only for linear models expressed as a formula. Any suggestions are welcome regarding plots or options to include. The first two plots are the same as the ones provided by `plot.lm`, the third is the dependant variable  $y$  with its mean  $\hat{y}$  (the fitted values) and the last plots the implied probabilities with the empirical density  $1/T$ .

### Examples

```

# GEL #
n = 500
phi<-c(.2, .7)
thet <- 0
sd <- .2
x <- matrix(arima.sim(n = n,list(order = c(2,0,1), ar = phi, ma = thet, sd = sd)), ncol = 1)
y <- x[7:n]
ym1 <- x[6:(n-1)]
ym2 <- x[5:(n-2)]

H <- cbind(x[4:(n-3)], x[3:(n-4)], x[2:(n-5)], x[1:(n-6)])

```

```

g <- y ~ ym1 + ym2
x <- H
t0 <- c(0, .5, .5)

res <- gel(g, x, t0)

plot(res, which = 3)
plot(res, which = 4)

# GMM #

res <- gmm(g, x)
plot(res, which = 3)

```

---

print

*Printing a gmm or gel object*


---

### Description

It is a printing method for `gmm` or `gel` objects.

### Usage

```

## S3 method for class 'gmm':
print(x, digits = 5, ...)
## S3 method for class 'gel':
print(x, digits = 5, ...)

```

### Arguments

<code>x</code>	An object of class <code>gmm</code> or <code>gel</code> returned by the function <code>gmm</code> or <code>gel</code>
<code>digits</code>	The number of digits to be printed
<code>...</code>	Other arguments when <code>print</code> is applied to an other class object

### Value

It prints some results from the estimation like the coefficients and the value of the objective function.

### Examples

```

# GMM #

n = 500
phi<-c(.2,.7)
thet <- 0
sd <- .2
x <- matrix(arima.sim(n = n, list(order = c(2,0,1), ar = phi, ma = thet, sd = sd)), ncol = 1)
y <- x[7:n]

```

```

ym1 <- x[6:(n-1)]
ym2 <- x[5:(n-2)]

H <- cbind(x[4:(n-3)], x[3:(n-4)], x[2:(n-5)], x[1:(n-6)])
g <- y ~ ym1 + ym2
x <- H

res <- gmm(g, x)
print(res)

# GEL #

t0 <- c(0, .5, .5)
res <- gel(g, x, t0)
print(res)

```

---

residuals

*Residuals of GEL or GMM*


---

## Description

Method to extract the residuals of the model estimated by `gmm` or `gel`.

## Usage

```

## S3 method for class 'gel':
residuals(object, ...)
## S3 method for class 'gmm':
residuals(object, ...)

```

## Arguments

`object` An object of class `gmm` or `gel` returned by the function `gmm` or `gel`  
`...` Other arguments when `residuals` is applied to an other classe object

## Value

It returns the matrix of residuals  $(y - \hat{y})$  in  $g=y \sim x$  as it is done by `residuals.lm`.

## Examples

```

# GEL can deal with endogeneity problems

n = 200
phi <- c(.2, .7)
thet <- 0.2
sd <- .2
set.seed(123)

```

```

x <- matrix(arima.sim(n = n, list(order = c(2,0,1), ar = phi, ma = thet, sd = sd)), ncol = 1)

y <- x[7:n]
ym1 <- x[6:(n-1)]
ym2 <- x[5:(n-2)]
H <- cbind(x[4:(n-3)], x[3:(n-4)], x[2:(n-5)], x[1:(n-6)])
g <- y ~ ym1 + ym2
x <- H

res <- gel(g, x, c(0,.3,.6))
e <- residuals(res)
plot(e, type = 'l', main = "Residuals from an ARMA fit using GEL")

# GMM is like GLS for linear models without endogeneity problems

set.seed(345)
n = 200
phi<-c(.2,.7)
thet <- 0
sd <- .2
x <- matrix(arima.sim(n = n, list(order = c(2,0,1), ar = phi, ma = thet, sd = sd)), ncol = 1)
y <- 10 + 5*rnorm(n) + x

res <- gmm(y ~ x, x)
plot(x, residuals(res), main = "Residuals of an estimated model with GMM")

```

rho

*Objective function of Generalized Empirical Likelihood (GEL)***Description**

It computes the objective function of GEL, its first and second analytical derivatives. It is called by `gel`.

**Usage**

```
rho(x, lamb, derive = 0, type = c("EL", "ET", "CUE"), drop = TRUE, k = 1)
```

**Arguments**

<code>x</code>	A $n \times q$ matrix with typical element $(t, i), g_i(\theta, x_t)$
<code>lamb</code>	A $q \times 1$ vector of lagrange multipliers
<code>derive</code>	0 for the objective function, 1 for the first derivative with respect to $\lambda$ and 2 for the second derivative with respect to $\lambda$ .
<code>type</code>	"EL" for empirical likelihood, "ET" for exponential tilting and "CUE" for continuous updated estimator.

drop	Because the solution may not be in the domain of $\rho(v) \forall t$ in small sample, we can drop those observations to avoid the return of NaN
k	It represents the ratio $k_1/k_2$ , where $k_1 = \int_{-\infty}^{\infty} k(s)ds$ and $k_2 = \int_{-\infty}^{\infty} k(s)^2 ds$ . See Smith(2004).

### Details

The objective function is  $\frac{1}{n} \sum_{t=1}^n \rho(\langle g(\theta, x_t), \lambda \rangle)$ , where  $\rho(v) = \log(1 - v)$  for empirical likelihood,  $-e^v$  for exponential tilting and  $(-v - 0.5v^2)$  for continuous updated estimator.

### Value

'rho' returns a scalar if "derive=0", a  $q$  vector if "derive" = 1 and a  $q \times q$  matrix if derive = 2.

### References

Newey, W.K. and Smith, R.J. (2004), Higher Order Properties of GMM and Generalized Empirical Likelihood Estimators. *Econometrica*, **72**, 219-255.

Hansen, L.P. and Heaton, J. and Yaron, A.(1996), Finit-Sample Properties of Some Alternative GMM Estimators. *Journal of Business and Economic Statistics*, **14** 262-280.

Smith, R.J. (2004), GEL Criteria for Moment Condition Models. *Working paper, CEMMAP*.

---

smoothG

*Kernel smoothing of a matrix of time series*

---

### Description

It applies the required kernel smoothing to the moment function in order for the GEL estimator to be valid. It is used by the gel function.

### Usage

```
smoothG(x, bw = bwAndrews2, prewhite = 1, ar.method = "ols", weights = weightsAndre
kernel = c("Bartlett", "Parzen", "Truncated", "Tukey-Hanning"),
approx = c("AR(1)", "ARMA(1,1)"), tol = 1e-7)
```

### Arguments

x	a $n \times q$ matrix of time series, where n is the sample size.
bw	The method to compute the bandwidth parameter. By default, it uses the bandwidth proposed by Andrews(1991). As an alternative, we can choose bw=bwNeweyWest2 (without "") which is proposed by Newey-West(1996).
prewhite	logical or integer. Should the estimating functions be prewhitened? If TRUE or greater than 0 a VAR model of order as.integer(prewhite) is fitted via ar with method "ols" and demean = FALSE.
ar.method	character. The method argument passed to ar for prewhitening.

weights	The smoothing weights can be computed by <code>weightsAndrews2</code> if it can be provided manually. If provided, it has to be a $r \times 1$ vector (see details).
approx	a character specifying the approximation method if the bandwidth has to be chosen by <code>bwAndrews2</code> .
tol	numeric. Weights that exceed <code>tol</code> are used for computing the covariance matrix, all other weights are treated as 0.
kernel	The choice of kernel

### Details

HAC is simply a modified version of `meatHAC` from the package `sandwich`. The modifications have been made so that the argument `x` can be a matrix instead of an object of class `lm` or `glm`. The details on how it works can be found on the `sandwich` manual.

The sample moment conditions  $\sum_{t=1}^n g(\theta, x_t)$  is replaced by:  $\sum_{t=1}^n g^k(\theta, x_t) = \sum_{i=-r}^r k(i)g(\theta, x_{t+i})$ , where  $r$  is a truncated parameter that depends on the bandwidth and  $k(i)$  are normalized weights so that they sum to 1.

If the vector of weights is provided, it gives only one side weights. For example, if you provide the vector  $(1, .5, .25)$ ,  $k(i)$  will become  $(.25, .5, 1, .5, .25) / (.25 + .5 + 1 + .5 + .25) = (.1, .2, .4, .2, .1)$

### Value

`smoothx`: A  $q \times q$  matrix containing an estimator of the asymptotic variance of  $\sqrt{n}\bar{x}$ , where  $\bar{x}$  is  $q \times 1$  vector with typical element  $\bar{x}_i = \frac{1}{n} \sum_{j=1}^n x_{ji}$ . This function is called by `gel` but can also be used by itself.

`kern_weights`: Vector of weights used for the smoothing.

### References

- Anatolyev, S. (2005), GMM, GEL, Serial Correlation, and Asymptotic Bias. *Econometrica*, **73**, 983-1002.
- Andrews DWK (1991), Heteroskedasticity and Autocorrelation Consistent Covariance Matrix Estimation. *Econometrica*, **59**, 817-858.
- Kitamura, Yuichi (1997), Empirical Likelihood Methods With Weakly Dependent Processes. *The Annals of Statistics*, **25**, 2084-2102.
- Zeileis A (2006), Object-oriented Computation of Sandwich Estimators. *Journal of Statistical Software*, **16**(9), 1-16. URL <http://www.jstatsoft.org/v16/i09/>.

### Examples

```
g <- function(tet, x)
{
  n <- nrow(x)
  u <- (x[7:n] - tet[1] - tet[2]*x[6:(n-1)] - tet[3]*x[5:(n-2)])
  f <- cbind(u, u*x[4:(n-3)], u*x[3:(n-4)], u*x[2:(n-5)], u*x[1:(n-6)])
  return(f)
}
n = 500
```

```

phi<-c(.2, .7)
thet <- 0.2
sd <- .2
x <- matrix(arima.sim(n = n, list(order = c(2, 0, 1), ar = phi, ma = thet, sd = sd)), ncol =
gt <- g(c(0, phi), x)
sgt <- smoothG(gt)$smoothx
plot(gt[,1])
lines(sgt[,1])

```

---

specTest

*Compute tests of specification*


---

### Description

Generic function for testing the specification of estimated models. It computes the J-test from `gmm` objects and J-test, LR-test and LM-test from `gel` objects.

### Usage

```

## S3 method for class 'gmm':
specTest(x, ...)
## S3 method for class 'gel':
specTest(x, ...)
## S3 method for class 'specTest':
print(x, digits = 5, ...)
specTest(x, ...)

```

### Arguments

<code>x</code>	A fitted model object.
<code>digits</code>	The number of digits to be printed.
<code>...</code>	Arguments passed to methods.

### Value

Tests and p-values

### References

Hansen, L.P. (1982), Large Sample Properties of Generalized Method of Moments Estimators. *Econometrica*, **50**, 1029-1054,

Smith, R. J. (2004), GEL Criteria for Moment Condition Models. *CeMMAP working papers, Institute for Fiscal Studies*

**Examples**

```
#####
n = 500
phi<-c(.2, .7)
thet <- 0
sd <- .2
x <- matrix(arima.sim(n=n,list (order=c(2,0,1),ar=phi,ma=thet,sd=sd)),ncol=1)
y <- x[7:n]
ym1 <- x[6:(n-1)]
ym2 <- x[5:(n-2)]

H <- cbind(x[4:(n-3)], x[3:(n-4)], x[2:(n-5)], x[1:(n-6)])
g <- y ~ ym1 + ym2
x <- H
t0 <- c(0, .5, .5)

res <- gel(g, x, t0)
specTest(res)

#####
res <- gmm(g, x)
specTest(res)
```

summary

*Method for object of class gmm or gel***Description**

It presents the results from the `gmm` or `gel` estimation in the same fashion as `summary` does for the `lm` class objects for example. It also compute the tests for overidentifying restrictions.

**Usage**

```
## S3 method for class 'gmm':
summary(object, ...)
## S3 method for class 'gel':
summary(object, ...)
## S3 method for class 'summary.gmm':
print(x, digits = 5, ...)
## S3 method for class 'summary.gel':
print(x, digits = 5, ...)
```

**Arguments**

`object` An object of class `gmm` or `gel` returned by the function `gmm` or `gel`

`x` An object of class `summary.gmm` or `summary.gel` returned by the function `summary.gmm` `summary.gel`

digits           The number of digits to be printed  
 ...             Other arguments when summary is applied to another class object

### Value

It returns a list with the parameter estimates and their standard deviations, t-stat and p-values. It also returns the J-test and p-value for the null hypothesis that  $E(g(\theta, X)) = 0$

### References

- Hansen, L.P. (1982), Large Sample Properties of Generalized Method of Moments Estimators. *Econometrica*, **50**, 1029-1054.
- Hansen, L.P. and Heaton, J. and Yaron, A.(1996), Finit-Sample Properties of Some Alternative GMM Estimators. *Journal of Business and Economic Statistics*, **14** 262-280.
- Anatolyev, S. (2005), GMM, GEL, Serial Correlation, and Asymptotic Bias. *Econometrica*, **73**, 983-1002.
- Kitamura, Yuichi (1997), Empirical Likelihood Methods With Weakly Dependent Processes. *The Annals of Statistics*, **25**, 2084-2102.
- Newey, W.K. and Smith, R.J. (2004), Higher Order Properties of GMM and Generalized Empirical Likelihood Estimators. *Econometrica*, **72**, 219-255.

### Examples

```
# GMM #

n = 500
phi<-c(.2, .7)
thet <- 0
sd <- .2
x <- matrix(arima.sim(n = n, list(order = c(2,0,1), ar = phi, ma = thet, sd = sd)), ncol = 1)
y <- x[7:n]
ym1 <- x[6:(n-1)]
ym2 <- x[5:(n-2)]

H <- cbind(x[4:(n-3)], x[3:(n-4)], x[2:(n-5)], x[1:(n-6)])
g <- y ~ ym1 + ym2
x <- H

res <- gmm(g, x)

summary(res)

# GEL #

t0 <- c(0, .5, .5)
res <- gel(g, x, t0)
summary(res)
```

vcov

*Variance-covariance matrix of GMM or GEL***Description**

It extracts the matrix of variances and covariances from `gmm` or `gel` objects.

**Usage**

```
## S3 method for class 'gmm':
vcov(object, ...)
## S3 method for class 'gel':
vcov(object, lambda = FALSE, ...)
```

**Arguments**

<code>object</code>	An object of class <code>gmm</code> or <code>gmm</code> returned by the function <code>gmm</code> or <code>gel</code>
<code>lambda</code>	If set to <code>TRUE</code> , the covariance matrix of the Lagrange multipliers is produced.
<code>...</code>	Other arguments when <code>vcov</code> is applied to another class object

**Value**

A matrix of variances and covariances

**Examples**

```
# GMM #
n = 500
phi<-c(.2,.7)
thet <- 0
sd <- .2
x <- matrix(arima.sim(n = n,list(order = c(2,0,1), ar = phi, ma = thet, sd = sd)), ncol = 1)
y <- x[7:n]
ym1 <- x[6:(n-1)]
ym2 <- x[5:(n-2)]

H <- cbind(x[4:(n-3)], x[3:(n-4)], x[2:(n-5)], x[1:(n-6)])
g <- y ~ ym1 + ym2
x <- H

res <- gmm(g, x)
vcov(res)

## GEL ##

t0 <- c(0,.5,.5)
res <- gel(g, x, t0)
vcov(res)
```

```
vcov(res, lambda = TRUE)
```

---

```
weightsAndrews2      Kernel weights
```

---

## Description

Function to compute the kernel weights used to compute the HAC covariance matrix

## Usage

```
weightsAndrews2(x, bw = bwAndrews2, kernel = c("Quadratic Spectral",
        "Truncated", "Bartlett", "Parzen", "Tukey-Hanning"), approx = c("AR(1)",
        "ARMA(1,1)"), prewhite = 1, ar.method = "ols", tol = 1e-7, verbose = FALSE)

bwAndrews2(x, kernel = c("Quadratic Spectral",
        "Truncated", "Bartlett", "Parzen", "Tukey-Hanning"), approx = c("AR(1)",
        "ARMA(1,1)"), prewhite = 1, ar.method = "ols")

bwNeweyWest2(x, kernel = c("Bartlett", "Parzen",
        "Quadratic Spectral", "Truncated", "Tukey-Hanning"),
        prewhite = 1, ar.method = "ols", ...)
```

## Arguments

<code>x</code>	A $n \times q$ matrix of time series from which we want to compute the covariance matrix.
<code>bw</code>	The method to compute the bandwidth parameter. For now, <code>bwAndrews2</code> is the only one possible. I leave the option there because I am planning to give more choices in futur versions of the package.
<code>prewhite</code>	logical or integer. Should the estimating functions be prewhitened? If <code>TRUE</code> or greater than 0 a VAR model of order <code>as.integer(prewhite)</code> is fitted via <code>ar</code> with method <code>"ols"</code> and <code>demean = FALSE</code> .
<code>ar.method</code>	character. The method argument passed to <code>ar</code> for prewhitening.
<code>verbose</code>	logical. Should the bandwidth parameter used be printed?
<code>approx</code>	a character specifying the approximation method if the bandwidth has to be chosen by <code>bwAndrews2</code> .
<code>tol</code>	numeric. Weights that exceed <code>tol</code> are used for computing the covariance matrix, all other weights are treated as 0.
<code>kernel</code>	The choice of kernel
<code>...</code>	It just allows to call either <code>bwAndrews2</code> or <code>bwNeweyWest</code> without having unusued arguments.

## Details

`weightsAndrews2`, `bwAndrews2` and `bwNeweyWest2` are simply modified version of `weightsAndrews`, `bwAndrews` and `bwNeweyWest` from the package `sandwich`. The modifications have been made so that the argument `x` can be a matrix instead of an object of class `lm` or `glm`. The details on how it works can be found on the `sandwich` manual. `kweights` is the same as the one included in the package `sandwich`.

## Value

`weightsAndrews` returns a vector of weights.

`bwAndrews` returns the selected bandwidth parameter.

## References

Zeileis A (2006), Object-oriented Computation of Sandwich Estimators. *Journal of Statistical Software*, **16**(9), 1–16. URL <http://www.jstatsoft.org/v16/i09/>.

Andrews DWK (1991), Heteroskedasticity and Autocorrelation Consistent Covariance Matrix Estimation. *Econometrica*, **59**, 817–858.

Newey WK & West KD (1987), A Simple, Positive Semi-Definite, Heteroskedasticity and Autocorrelation Consistent Covariance Matrix. *Econometrica*, **55**, 703–708.

Newey WK & West KD (1994), Automatic Lag Selection in Covariance Matrix Estimation. *Review of Economic Studies*, **61**, 631-653.

## Examples

```
set.seed(345)
x <- arima.sim(n = 200, list(ordre = c(1, 0, 1), ar = .9, ma = .4))
y <- 2*x + rnorm(200)
x = cbind(x, y)
w <- weightsAndrews2(x, bw = bwAndrews2, kernel = "Quadratic")
plot(w, type = 'l')
w2 <- weightsAndrews2(x, bw = bwNeweyWest2, kernel = "Bartlett")
plot(w2, type = 'l')
```

# Index

## \*Topic **datasets**

Finance, 5

## \*Topic **regression**

kweights2, 22

## \*Topic **ts**

kweights2, 22

ar, 10, 17, 21, 29, 34

bwAndrews2, 10, 11, 17, 21

bwAndrews2 (weightsAndrews2), 34

bwNeweyWest2, 10, 17, 21

bwNeweyWest2 (weightsAndrews2), 34

charStable, 2

coef, 3

confint, 4

constrOptim, 11, 12

Finance, 5

FinRes, 6

fitted, 7

formula, 8

gel, 3, 4, 7, 9, 11, 13, 24–27, 29, 32, 33

getDat, 13

getLamb, 10–12, 14

getModel, 16

gmm, 3, 4, 7, 9, 13, 16, 18, 23–26, 32, 33

HAC, 10, 17, 21

kweights (weightsAndrews2), 34

kweights2, 22

lm, 11, 18

momentEstim, 23

nlminb, 11, 18

optim, 11, 12, 17, 18

optimize, 11, 12, 17

panel.smooth, 24

par, 24

plot, 24

points, 24

print, 25

print.specTest (specTest), 30

print.summary.gel (summary), 31

print.summary.gmm (summary), 31

residuals, 26

rho, 27

smoothG, 28

specTest, 30

summary, 31

summary.gel, 32

summary.gel (summary), 31

summary.gmm, 32

summary.gmm (summary), 31

terms, 12, 19

vcov, 33

weightsAndrews2, 11, 29, 34