

Package: ivx (via r-universe)

September 23, 2026

Type Package

Title Robust Econometric Inference

Version 2.0.0

Description Drawing statistical inference on the coefficients of a short- or long-horizon predictive regression with persistent regressors by using the IVX method of Magdalinos and Phillips (2009) <[doi:10.1017/S0266466608090154](https://doi.org/10.1017/S0266466608090154)> and Kostakis, Magdalinos and Stamatogiannis (2015) <[doi:10.1093/rfs/hhu139](https://doi.org/10.1093/rfs/hhu139)>. Also provides the extensions of IVX proposed in the later literature (IVX-AR, residual-augmented and lag-augmented IVX, IVX quantile regression, systems of predictive regressions, heteroskedasticity-robust standard errors, wild-bootstrap and subsample inference, combined instruments) and the standard non-IVX tests for predictability with persistent regressors as benchmarks.

License GPL-3

URL <https://kvasilopoulos.github.io/ivx/>,
<https://github.com/kvasilopoulos/ivx>

BugReports <https://github.com/kvasilopoulos/ivx/issues>

Depends R (>= 3.5)

Imports methods, Rcpp (>= 0.12.18)

Suggests covr (>= 3.2.1), forecast (>= 8.12), spelling (>= 2.1),
testthat (>= 2.1.1), lmtest, knitr, rmarkdown, quantreg

Enhances texreg

VignetteBuilder knitr

LinkingTo Rcpp (>= 1.0.1), RcppArmadillo (>= 0.9.300.2.0)

Encoding UTF-8

Language en-GB

LazyData true

Roxygen list(markdown = TRUE)

Config/roxygen2/version 8.1.0

Repository <https://kvasilopoulos.r-universe.dev>

Date/Publication 2026-09-23 08:35:56 UTC

RemoteUrl <https://github.com/kvasilopoulos/ivx>

RemoteRef HEAD

RemoteSha 533492e3dea143449f1e69e65e3f81d035fc2693

Contents

ac_test	3
ac_test_	3
arm	4
arm_fit	6
cy_test	7
delta	8
el_test	9
elliott_cf	10
extract.ivx	12
hlt_test	13
ivx	15
ivx_ar	17
ivx_ar_fit	19
ivx_boot	20
ivx_episodic	22
ivx_fit	23
ivx_iv	25
ivx_iv_fit	27
ivx_qr	27
ivx_qr_boot	29
ivx_qr_fit	30
ivx_ra	31
ivx_ra_fit	33
ivx_sys	34
ivx_sys_fit	35
kms	36
kms_quarterly	37
summary.ivx	38
summary.ivx_ar	38
vcov.ivx	39
ylpc	40

Index	41
--------------	-----------

ac_test	<i>Autocorrelation tests</i>
---------	------------------------------

Description

Autocorrelation tests

Usage

```
ac_test(x, lag_max = 5)
```

Arguments

x	the residuals or an ivx object.
lag_max	the maximum length of lags.

Examples

```
obj <- ivx(hpi ~ cpi + def + int + log(res), data = ylpc)
if (requireNamespace("lmtest", quietly = TRUE)) {
  lmtest::bgtest(hpi ~ cpi + def + int + log(res), data = ylpc)
}
ac_test(obj, 5)
```

ac_test_	<i>Tests for autocorrelation</i>
----------	----------------------------------

Description

- ac_test: For all tests
- ac_test_wald: Wald test
- ac_test_lb: Ljung-Box
- ac_test_bp: Box-Pierce
- ac_test_bg: Breusch-Godfrey

Usage

```
ac_test_wald(x, lag)

ac_test_lb(x, lag)

ac_test_bp(x, lag)

ac_test_bg(x, order, type, fill)
```

Arguments

<code>x</code>	an ivx model or a numeric vector, usually the residuals from an ols regression.
<code>lag</code>	the number of lags.
<code>order</code>	lag TODO
<code>type</code>	the type of test statistic to be returned. Either "Chisq" for the Chi-squared test statistic or "F" for the F test statistic.
<code>fill</code>	starting values for the lagged residuals in the auxiliary regression. By default 0 but can also be set to NA.

Details

If $p\text{-value} < 0.051$: You can reject the null hypothesis assuming a 5% chance of making a mistake. So you can assume that your values are showing dependence on each other.

Value

a numeric scalar or numeric vector.

See Also

`Box.test lmtest::bgtest`

Examples

```
mdl <- ivx(hpi ~ cpi + inv, data = y1pc)
ac_test_wald(mdl)

ac_test(mdl)
```

arm

Augmented Regression Method (Amihud, Hurvich & Wang)

Description

`arm` implements the multipredictor augmented regression method (mARM) of Amihud, Hurvich and Wang (2009), a reduced-bias OLS alternative to IVX for stationary but persistent predictors. A VAR(1) is fitted to the predictors, its coefficient matrix is bias-corrected with the Nicholls and Pope (1988) expansion (iterated), and the predictive regression is augmented with the corrected VAR residuals, which removes the Stambaugh (1999) bias from the slopes. Standard errors and the joint Wald statistic use the covariance estimator of the paper (eqs 7-8), which adds the estimation uncertainty of the VAR coefficients to the augmented-regression OLS variance.

Usage

```

arm(
  formula,
  data,
  iter = 10,
  na.action,
  contrasts = NULL,
  model = TRUE,
  x = FALSE,
  y = FALSE,
  ...
)

## S3 method for class 'arm'
print(x, digits = max(3L, getOption("digits") - 3L), ...)

```

Arguments

formula	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
data	n optional data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which lm is called.
iter	maximum number of bias-correction iterations ($K = 10$ in the paper); iteration stops earlier if the corrected VAR becomes non-stationary.
na.action	a function which indicates what should happen when the data contain NAs. The default is set by the na.action setting of options , and is na.fail if that is unset. The ‘factory-fresh’ default is na.omit . Another possible value is NULL, no action. Value na.exclude can be useful.
contrasts	an optional list. See the contrasts.arg of model.matrix.default .
model	logical. If TRUE the model.frame of the fit is returned.
x	an object of class "arm".
y	logical. If TRUE the response of the fit is returned.
...	additional arguments to be passed to the low level regression fitting functions (see lm).
digits	the number of significant digits to use when printing.

Details

Unlike IVX the method assumes stationary predictors (all eigenvalues of the VAR coefficient matrix inside the unit circle) and Gaussian innovations; it is the natural benchmark for the "control function" approach of Elliott (2011). Short horizon only.

Value

an object of class `c("arm", "ivx")`, so the `ivx` methods apply. Additional components: `phi` (coefficients on the augmentation residuals), `Phi` (bias-corrected VAR(1) coefficient matrix, equations by row) and `Phi_ols`.

References

Amihud, Y., Hurvich, C. M., & Wang, Y. (2009). Multiple-predictor regressions: Hypothesis testing. *The Review of Financial Studies*, 22(1), 413-434.

Amihud, Y., & Hurvich, C. M. (2004). Predictive regressions: A reduced-bias estimation method. *Journal of Financial and Quantitative Analysis*, 39(4), 813-841.

Nicholls, D. F., & Pope, A. L. (1988). Bias in the estimation of multivariate autoregressions. *Australian Journal of Statistics*, 30A, 296-309.

Examples

```
arm(Ret ~ DP, data = kms)

summary(arm(Ret ~ DP + TBL, data = kms))
```

arm_fit	<i>Fitter Function for the Augmented Regression Method</i>
---------	--

Description

Basic function called by `arm`. Should only be used directly by experienced users.

Usage

```
arm_fit(y, x, iter = 10, ...)
```

Arguments

<code>y</code>	vector of observations of length <code>n</code> , or a matrix with <code>n</code> rows.
<code>x</code>	design matrix of dimension <code>n * p</code> .
<code>iter</code>	maximum number of bias-correction iterations (<code>K = 10</code> in the paper); iteration stops earlier if the corrected VAR becomes non-stationary.
<code>...</code>	currently unused.

Examples

```
arm_fit(kms$Ret, as.matrix(kms$DP))$tstat
```

Description

cy_test implements the Bonferroni Q-test of Campbell and Yogo (2006), the standard feasible version of the sup-bound / Bonferroni approach of Cavanagh, Elliott and Stock (1995). For a single predictor $x_t = \gamma + \rho x_{t-1} + v_t$ (AR(p) dynamics allowed, Appendix A), the procedure is:

1. estimate the innovation correlation δ between the predictive-regression residual and the ADF innovation of the predictor (lag length by BIC, $p \in [1, p_{max}]$);
2. compute the DF-GLS statistic of Elliott, Rothenberg and Stock (1996) and invert its local-to-unity null distribution (Stock, 1991) into a confidence interval $[\underline{c}, \bar{c}]$ for $c = T(\rho - 1)$, with lower and upper levels $\underline{\alpha}_1(\delta)$, $\bar{\alpha}_1(\delta)$ from the paper's Table 2, which refine the Bonferroni bound so that the one-sided test has size 5\
3. for ρ at each end of the interval compute the Q-estimate $\hat{\beta}(\rho)$ of eq. (25) - OLS of y_t on the demeaned x_{t-1} after subtracting $(\sigma_{ue}/\sigma_e\omega)(x_t - \rho x_{t-1})$, with the Phillips-Perron-type correction $\frac{T}{2}(\sigma_{ue}/\sigma_e\omega)(\omega^2 - \sigma_v^2)$ when $p > 1$ - and its standard error $\sigma_u(1 - \delta^2)^{1/2}/(\sum x_{t-1}^2)^{1/2}$;
4. the 90\ $[\hat{\beta}(\bar{\rho}) - 1.645 se, \hat{\beta}(\underline{\rho}) + 1.645 se]$ (eq. 17); the 5\ bound is positive and $\beta \geq 0$ if the upper bound is negative.

The DF-GLS null quantiles are tabulated by simulation for $c \in [-100, 10]$ (see data-raw/dfgls-quantiles.R); a statistic outside the tabulated range is clamped to the boundary, which for very negative values (a clearly stationary predictor) makes the interval for c start at -100 . Table 2 is given for $\delta < 0$; for $\hat{\delta} > 0$ the predictor is sign-flipped, which flips β and the alternative, and the results are mapped back.

Usage

```
cy_test(formula, data, lag_max = NULL, na.action)

cy_test_fit(y, x, lag_max = NULL)

## S3 method for class 'cy_test'
print(x, digits = max(3L, getOption("digits") - 3L), ...)
```

Arguments

formula	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
data	n optional data frame, list or environment (or object coercible by <code>as.data.frame</code> to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which lm is called.
lag_max	maximum ADF lag order for the BIC search; the default is $\lfloor 12(T/100)^{1/4} \rfloor$ lagged differences.

na.action	a function which indicates what should happen when the data contain NAs. The default is set by the na.action setting of <code>options</code> , and is <code>na.fail</code> if that is unset. The ‘factory-fresh’ default is <code>na.omit</code> . Another possible value is NULL, no action. Value <code>na.exclude</code> can be useful.
y	response vector.
x	an object of class "cy_test".
digits	minimal number of significant digits.
...	unused.

Value

an object of class "cy_test": a list with ci (the 90\ interval for β), reject (named logical: greater, less), estimate (OLS slope), beta_rho ($\hat{\beta}$ at $\underline{\rho}$ and $\bar{\rho}$), se, delta, dfgls, c_ci, rho_ci, alpha1, lag and n.

References

- Campbell, J. Y., & Yogo, M. (2006). Efficient tests of stock return predictability. *Journal of Financial Economics*, 81(1), 27-60.
- Cavanagh, C. L., Elliott, G., & Stock, J. H. (1995). Inference in models with nearly integrated regressors. *Econometric Theory*, 11(5), 1131-1147.
- Elliott, G., Rothenberg, T. J., & Stock, J. H. (1996). Efficient tests for an autoregressive unit root. *Econometrica*, 64(4), 813-836.
- Stock, J. H. (1991). Confidence intervals for the largest autoregressive root in U.S. macroeconomic time series. *Journal of Monetary Economics*, 28(3), 435-459.

Examples

```
cy_test(Ret ~ DP, data = kms)
```

delta	<i>Calculate the delta coefficient</i>
-------	--

Description

Computes the long-run correlation coefficient between the residuals of the predictive regression and the autoregressive model for the regressor.

Usage

```
delta(object)
```

Arguments

object on object of class "ivx"

Value

A vector of the estimated correlation coefficients. This should have row and column names corresponding to the parameter names given by the `coef` method.

Examples

```
mod <- ivx(Ret ~ LTY, data = monthly)

delta(mod)
```

el_test	<i>Unified Empirical Likelihood Test for Predictability (Liu, Yang, Cai & Peng)</i>
---------	---

Description

`el_test` implements the unified empirical likelihood (EL) test of Liu, Yang, Cai and Peng (2019, Section 2.2) for the predictive regression with unknown intercept

$$Y_t = \alpha + \beta_1 \Delta X_{t-1} + \beta_2 X_{t-2} + U_t,$$

where the lagged difference of the predictor is included so that the response can be stationary when the predictor is not. The intercept is removed by differencing at lag $m = \lfloor n/2 \rfloor$ ($\tilde{Y}_t = Y_{t+m} - Y_t$, $\tilde{X}_t = X_{t+m} - X_t$), and the EL function is built from the score equations $\tilde{Z}_{t1} = \tilde{e}_t \Delta \tilde{X}_{t-1}$ and $\tilde{Z}_{t2} = \tilde{e}_t \tilde{X}_{t-2} / \sqrt{1 + \tilde{X}_{t-2}^2}$, $t = 3, \dots, m$, with $\tilde{e}_t$ the model error. The weight on the second equation makes its sample variance converge whether the predictor is stationary, nearly integrated or a unit root, so that the profile EL ratios for $H_0 : \beta_2 = 0$ (no predictability), $H_0 : \beta_1 = 0$ and the joint null are $\chi^2(1)$, $\chi^2(1)$ and $\chi^2(2)$ without knowing the persistence (Theorem 2). Single predictor only.

Usage

```
el_test(formula, data, na.action)

el_test_fit(y, x)

## S3 method for class 'el_test'
print(x, digits = max(3L, getOption("digits") - 3L), ...)
```

Arguments

formula	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
data	n optional data frame, list or environment (or object coercible by <code>as.data.frame</code> to a data frame) containing the variables in the model. If not found in data, the variables are taken from <code>environment(formula)</code> , typically the environment from which <code>lm</code> is called.

na.action	a function which indicates what should happen when the data contain NAs. The default is set by the na.action setting of <code>options</code> , and is <code>na.fail</code> if that is unset. The ‘factory-fresh’ default is <code>na.omit</code> . Another possible value is NULL, no action. Value <code>na.exclude</code> can be useful.
y	response vector.
x	an object of class "el_test".
digits	minimal number of significant digits.
...	unused.

Value

an object of class "el_test": a list with the EL ratio statistics `stat` (named `beta2`, `beta1`, `joint`), their `p.value`, the unconstrained EL estimates `estimate`, the OLS estimates `ols` of (β_1, β_2) on the differenced data, and `m`.

References

Liu, X., Yang, B., Cai, Z., & Peng, L. (2019). A unified test for predictability of asset returns regardless of properties of predicting variables. *Journal of Econometrics*, 208(1), 141-159.

Owen, A. B. (2001). *Empirical Likelihood*. Chapman & Hall.

Examples

```
el_test(Ret ~ DP, data = kms)
```

elliott_cf

Control-Function Predictability Test of Elliott (2011)

Description

`elliott_cf` runs the augmented predictive regression of Elliott (2011, eq. 9),

$$y_t = \alpha + \beta' x_{t-1} + \gamma' Z_t + \tilde{u}_t, \quad Z_t = (z'_t, z'_{t-1}, \dots, z'_{t-q})',$$

where z_t are user-supplied stationary covariates that are contemporaneously correlated with the innovations of the persistent predictors x_t and of the response. If they absorb that correlation (the "orthogonalising" condition, Section 5 of the paper), the Wald test of $\beta = 0$ has a standard chi-square limit whatever the persistence of x_t (Theorem 2); without them it has the non-standard Elliott-Stock (1994) distribution (Theorem 1). Unlike `arm()` and `ivx_ra()`, which build the control variable from the data, the covariates here come from the user

- the paper's example is predicting returns with the dividend-price ratio using contemporaneous price-related variables.

Usage

```
elliott_cf(formula, covariates, data, lags = 0, robust = TRUE, na.action)

elliott_cf_fit(y, x, z, lags = 0, robust = TRUE)

## S3 method for class 'elliott_cf'
print(x, digits = max(3L, getOption("digits") - 3L), ...)
```

Arguments

formula	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
covariates	a one-sided formula giving the orthogonalising covariates z_t (contemporaneous with the response).
data	n optional data frame, list or environment (or object coercible by <code>as.data.frame</code> to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which lm is called.
lags	number of lags q of the covariates to include.
robust	logical; if TRUE (default) Eicker-White standard errors are used.
na.action	a function which indicates what should happen when the data contain NAs. The default is set by the na.action setting of <code>options</code> , and is <code>na.fail</code> if that is unset. The ‘factory-fresh’ default is <code>na.omit</code> . Another possible value is NULL, no action. Value <code>na.exclude</code> can be useful.
y	response vector.
x	an object of class "elliott_cf".
z	matrix of covariates.
digits	minimal number of significant digits.
...	unused.

Details

The remaining innovation correlation after augmentation is returned as a diagnostic: it is the correlation between the regression residuals and the residuals of an AR(1) of each predictor on the same covariates, and should be close to zero for the test to be reliable.

Value

an object of class "elliott_cf": a list with coefficients (on the predictors), se, tstat, Wald, df, p.value, gamma (coefficients on the covariates and their lags), rho_resid (remaining innovation correlation per predictor) and the underlying lm fit.

References

Elliott, G. (2011). A control function approach for testing the usefulness of trending variables in predictive regressions and econometric models. *Journal of Econometrics*, 164(1), 79-91.

Elliott, G., & Stock, J. H. (1994). Inference in time series regression when the order of integration of a regressor is unknown. *Econometric Theory*, 10(3-4), 672-700.

Examples

```
# the T-bill rate as covariate for the dividend-price ratio (illustration only)
elliott_cf(Ret ~ DP, ~ TBL, data = kms)
```

extract.ivx	extract <i>method for ivx objects</i>
-------------	---------------------------------------

Description

extract method for ivx objects

Usage

```
extract.ivx(
  model,
  include.wald = TRUE,
  include.nobs = TRUE,
  include.aic = FALSE,
  include.bic = FALSE,
  include.rsquared = FALSE,
  include.adjrs = FALSE,
  ...
)

extract.ivx_ar(
  model,
  include.wald = TRUE,
  include.nobs = TRUE,
  include.aic = FALSE,
  include.bic = FALSE,
  include.rsquared = FALSE,
  include.adjrs = FALSE,
  ...
)
```

Arguments

model	A statistical model object.
include.wald	Report the Wald statistic.

include.nobs	Report the number of observations in the GOF block?
include.aic	Report Akaike's Information Criterion (AIC) in the GOF block?
include.bic	Report the Bayesian Information Criterion (BIC) in the GOF block?
include.rsquared	Report the R-squared.
include.adjrs	Report the Adjusted R-squared.
...	Custom parameters, which are handed over to subroutines. Currently not in use.

hlt_test	<i>Hybrid t-Test for Return Predictability (Harvey, Leybourne & Taylor)</i>
----------	---

Description

`hlt_test` implements the double-switching hybrid procedure T_{hyb} of Harvey, Leybourne and Taylor (2021, Section 3.3) for a single predictor. Two regression t-ratios are used: the standard OLS t-ratio T (eq. 5) and the variant $\tilde{T}$ in which the predictor is quasi-GLS demeaned with $\bar{c} = 7$ (eq. 7). Under weak persistence the standard t-ratio is compared with normal critical values; under strong persistence the limiting null distributions depend on the local-to-unity parameter and on the innovation correlation ρ_{xy} , and the tests are run with the paper's asymptotically conservative critical values (maximised over c) obtained from the response surfaces in its Table 1.

Usage

```
hlt_test(
  formula,
  data,
  alternative = c("greater", "less"),
  level = 0.05,
  lag_max = NULL,
  na.action
)

hlt_test_fit(y, x, alternative = "greater", level = 0.05, lag_max = NULL)

## S3 method for class 'hlt_test'
print(x, digits = max(3L, getOption("digits") - 3L), ...)
```

Arguments

formula	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
data	n optional data frame, list or environment (or object coercible by <code>as.data.frame</code> to a data frame) containing the variables in the model. If not found in data, the variables are taken from <code>environment(formula)</code> , typically the environment from which <code>lm</code> is called.

alternative	direction of the one-sided test.
level	significance level; one of 0.1, 0.05, 0.025, 0.01 (the levels for which response surfaces are available).
lag_max	maximum ADF lag order; the default is the paper's $\lfloor 12(T/100)^{1/4} \rfloor$.
na.action	a function which indicates what should happen when the data contain NAs. The default is set by the na.action setting of <code>options</code> , and is <code>na.fail</code> if that is unset. The 'factory-fresh' default is <code>na.omit</code> . Another possible value is <code>NULL</code> , no action. Value <code>na.exclude</code> can be useful.
y	response vector.
x	an object of class "hlt_test".
digits	minimal number of significant digits.
...	unused.

Details

The procedure is: (1) if the ADF normalised-bias statistic $T\hat{\rho}/(1 - \sum_i \hat{\psi}_i)$ from an ADF regression with lag length chosen by the MBIC of Ng and Perron (2001) is below $-4\sqrt{T}$, the predictor is treated as weakly persistent and the standard test T_N is used; (2) otherwise, for an upper-tail test, T with critical value $cv(\hat{\rho}_{xy})$ if $\hat{\rho}_{xy} > -0.1$ and $\tilde{T}$ with $\tilde{cv}(\hat{\rho}_{xy})$ if $\hat{\rho}_{xy} < -0.1$ (mirrored for lower-tail tests), where $\hat{\rho}_{xy}$ is the correlation of the ADF residuals with the predictive-regression residuals.

Value

an object of class "hlt_test": a list with the selected test ("T_N", "T_con" or "T~_con"), its statistic, cv and reject indicator, plus t, t_gls (both t-ratios), adf, adf_lag, rho_xy and estimate (the OLS slope).

References

Harvey, D. I., Leybourne, S. J., & Taylor, A. M. R. (2021). Simple tests for stock return predictability with good size and power properties. *Journal of Econometrics*, 224(1), 198-214.

Ng, S., & Perron, P. (2001). Lag length selection and the construction of unit root tests with good size and power. *Econometrica*, 69(6), 1519-1554.

Examples

```
hlt_test(Ret ~ DP, data = kms)
hlt_test(Ret ~ TBL, data = kms, alternative = "less", level = 0.1)
```

ivx

*Fitting IVX Models***Description**

ivx fits predictive regression models. The method allows standard chi-square testing for regressors with different degrees of persistence, from stationary to mildly explosive, and can be used for both short- and long-horizon predictive regressions.

Usage

```
ivx(
  formula,
  data,
  horizon,
  na.action,
  weights,
  contrasts = NULL,
  offset,
  model = TRUE,
  x = FALSE,
  y = FALSE,
  beta = 0.95,
  cz = 1,
  bandwidth = NULL,
  robust = FALSE,
  lag_y = FALSE,
  ...
)

## S3 method for class 'ivx'
print(x, digits = max(3L, getOption("digits") - 3L), ...)
```

Arguments

formula	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
data	n optional data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which lm is called.
horizon	is the horizon (default horizon = 1 corresponds to a short-horizon regression). For horizon > 1 the estimator and modified Wald statistic are those of Kostakis et al. (2023), eqs (15) and (23): K-period sums of the response and of the lagged predictors, a single-lag instrument in the signal matrix and the K-period sum of the instrument in the covariance.

<code>na.action</code>	a function which indicates what should happen when the data contain NAs. The default is set by the <code>na.action</code> setting of <code>options</code> , and is <code>na.fail</code> if that is unset. The ‘factory-fresh’ default is <code>na.omit</code> . Another possible value is <code>NULL</code> , no action. Value <code>na.exclude</code> can be useful.
<code>weights</code>	an optional vector of weights to be used in the fitting process. Should be <code>NULL</code> or a numeric vector. If non- <code>NULL</code> , weighted least squares is used with weights <code>weights</code> (that is, minimizing $\sum(w \cdot e^2)$); otherwise ordinary least squares is used.
<code>contrasts</code>	an optional list. See the <code>contrasts.arg</code> of <code>model.matrix.default</code> .
<code>offset</code>	this can be used to specify an a priori known component to be included in the linear predictor during fitting. This should be <code>NULL</code> or a numeric vector or matrix of extents matching those of the response. One or more offset terms can be included in the formula instead or as well, and if more than one are specified their sum is used. See <code>model.offset</code>
<code>model</code>	logical. If <code>TRUE</code> the <code>model.frame</code> of the fit is returned.
<code>x</code>	an object of class "ivx", usually, a result of a call to <code>ivx</code> .
<code>y</code>	logical. If <code>TRUE</code> the response of the fit is returned.
<code>beta, cz</code>	tuning parameters of the IVX instrument $z_t = \sum_{j=0}^{t-1} (1 - c_z/n^\beta)^j \Delta x_{t-j}$. Defaults (<code>beta = 0.95</code> , <code>cz = 1</code>) follow Kostakis et al. (2015).
<code>bandwidth</code>	Newey-West bandwidth for the long-run covariance estimate. The default <code>NULL</code> uses $n^{1/3}$ as in Kostakis et al. (2015).
<code>robust</code>	logical. If <code>TRUE</code> the Eicker-White (heteroskedasticity-robust) form of the IVX covariance matrix is used (Demetrescu et al., 2023). Only available for <code>horizon = 1</code> .
<code>lag_y</code>	logical. If <code>TRUE</code> the regression is augmented with the lagged dependent variable (column <code>y_lag</code>), instrumented by itself, as in Demetrescu (2014): this can raise the local power of the IVX test when the predictors are highly persistent and endogenous, at no cost otherwise. The joint Wald statistic still tests only the predictors. Only for <code>horizon = 1</code> .
<code>...</code>	additional arguments to be passed to the low level regression fitting functions (see <code>lm</code>).
<code>digits</code>	the number of significant digits to use when printing.

Value

an object of class "ivx".

References

- Magdalinos, T., & Phillips, P. (2009). Limit Theory for Cointegrated Systems with Moderately Integrated and Moderately Explosive Regressors. *Econometric Theory*, 25(2), 482-526.
- Kostakis, A., Magdalinos, T., & Stamatogiannis, M. P. (2015). Robust econometric inference for stock return predictability. *The Review of Financial Studies*, 28(5), 1506-1553.
- Demetrescu, M., Georgiev, I., Rodrigues, P. M. M., & Taylor, A. M. R. (2023). Extensions to IVX methods of inference for return predictability. *Journal of Econometrics*, 237(2), 105271.

Kostakis, A., Magdalinos, T., & Stamatogiannis, M. P. (2023). Taking stock of long-horizon predictability tests: Are factor returns predictable? *Journal of Econometrics*, 237(2), 105380.

Demetrescu, M. (2014). Enhancing the local power of IVX-based tests in predictive regressions. *Economics Letters*, 124(2), 269-273.

Examples

```
# Univariate
ivx(Ret ~ LTY, data = kms)

# Multivariate
ivx(Ret ~ LTY + TBL, data = kms)

# Longer horizon
ivx(Ret ~ LTY + TBL, data = kms, horizon = 4)

wt <- runif(nrow(kms))
ivx(Ret ~ LTY, data = kms, weights = wt)

# lag-augmented IVX (Demetrescu, 2014)
ivx(Ret ~ DP, data = kms, lag_y = TRUE)
```

ivx_ar

Fitting IVX-AR Models

Description

ivx_ar implements the Yang et al (2020) new instrumental variable based Wald statistic (IVX-AR) which accounts for serial correlation and heteroscedasticity in the error terms of the linear predictive regression model.

Usage

```
ivx_ar(
  formula,
  data,
  horizon,
  ar = "auto",
  ar_ic = c("bic", "aic", "aicc"),
  ar_max = 5,
  ar_grid = function(x) seq(x - 0.3, x + 0.3, by = 0.02),
  na.action,
  contrasts = NULL,
  offset,
  model = TRUE,
  x = FALSE,
  y = FALSE,
```

```

    beta = 0.95,
    cz = 1,
    bandwidth = NULL,
    robust = FALSE,
    ...
)

## S3 method for class 'ivx_ar'
print(x, digits = max(3L, getOption("digits") - 3L), ...)

```

Arguments

formula	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
data	n optional data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which lm is called.
horizon	is the horizon (default horizon = 1 corresponds to a short-horizon regression). For horizon > 1 the estimator and modified Wald statistic are those of Kostakis et al. (2023), eqs (15) and (23): K-period sums of the response and of the lagged predictors, a single-lag instrument in the signal matrix and the K-period sum of the instrument in the covariance.
ar	Method to include the autoregressive terms. "auto" find the optimal ar order by using the information criteria. ar = 0 reduces to simple ivx . ar > 1 uses a fixed order to estimate the model.
ar_ic	Information criterion to be used in model selection.
ar_max	Maximum ar order of model to fit.
ar_grid	The ar grid sequence of which to iterate.
na.action	a function which indicates what should happen when the data contain NAs. The default is set by the na.action setting of options , and is na.fail if that is unset. The 'factory-fresh' default is na.omit . Another possible value is NULL, no action. Value na.exclude can be useful.
contrasts	an optional list. See the contrasts.arg of model.matrix.default .
offset	this can be used to specify an a priori known component to be included in the linear predictor during fitting. This should be NULL or a numeric vector or matrix of extents matching those of the response. One or more offset terms can be included in the formula instead or as well, and if more than one are specified their sum is used. See model.offset
model	logical. If TRUE the model.frame of the fit is returned.
x	an object of class "ivx_ar", usually, a result of a call to ivx_ar .
y	logical. If TRUE the response of the fit is returned.
beta, cz	tuning parameters of the IVX instrument $z_t = \sum_{j=0}^{t-1} (1 - c_z/n^\beta)^j \Delta x_{t-j}$. Defaults (beta = 0.95, cz = 1) follow Kostakis et al. (2015).

bandwidth	Newey-West bandwidth for the long-run covariance estimate. The default NULL uses $n^{1/3}$ as in Kostakis et al. (2015).
robust	logical. If TRUE the Eicker-White (heteroskedasticity-robust) form of the IVX covariance matrix is used (Demetrescu et al., 2023). Only available for horizon = 1.
...	additional arguments to be passed to the low level regression fitting functions (see lm).
digits	the number of significant digits to use when printing.

References

Yang, B., Long, W., Peng, L., & Cai, Z. (2020). Testing the Predictability of US Housing Price Index Returns Based on an IVX-AR Model. *Journal of the American Statistical Association*, 1-22. DOI: 10.1080/01621459.2019.1686392

Examples

```
ivx_ar(hpi ~ log(res) + cpi, ylpc)

ivx_ar(hpi ~ log(res) + cpi, ylpc, ar_ic = "aic")

ivx_ar(hpi ~ log(res) + cpi, ylpc, ar = 1)
```

ivx_ar_fit

Fitter Functions for IVX-AR Models

Description

Basic function called by ivx_ar to fit predictive models. These should only be used directly by experienced users.

Usage

```
ivx_ar_fit(
  y,
  x,
  horizon = 1,
  offset = NULL,
  ar = "auto",
  ar_max = 5,
  ar_ic = "bic",
  ar_grid = function(x) seq(x - 0.3, x + 0.3, by = 0.02),
  beta = 0.95,
  cz = 1,
  bandwidth = NULL,
  robust = FALSE,
  ...
)
```

Arguments

y	vector of observations of length n, or a matrix with n rows.
x	design matrix of dimension $n \times p$.
horizon	is the horizon (default horizon = 1 corresponds to a short-horizon regression). For horizon > 1 the estimator and modified Wald statistic are those of Kostakis et al. (2023), eqs (15) and (23): K-period sums of the response and of the lagged predictors, a single-lag instrument in the signal matrix and the K-period sum of the instrument in the covariance.
offset	(numeric of length n). This can be used to specify an <i>a priori</i> known component to be included in the linear predictor during fitting.
ar	Method to include the autoregressive terms. "auto" find the optimal ar order by using the information criteria. ar = 0 reduces to simple ivx . ar > 1 uses a fixed order to estimate the model.
ar_max	Maximum ar order of model to fit.
ar_ic	Information criterion to be used in model selection.
ar_grid	The ar grid sequence of which to iterate.
beta, cz	tuning parameters of the IVX instrument $z_t = \sum_{j=0}^{t-1} (1 - c_z/n^\beta)^j \Delta x_{t-j}$. Defaults (beta = 0.95, cz = 1) follow Kostakis et al. (2015).
bandwidth	Newey-West bandwidth for the long-run covariance estimate. The default NULL uses $n^{1/3}$ as in Kostakis et al. (2015).
robust	logical. If TRUE the Eicker-White (heteroskedasticity-robust) form of the IVX covariance matrix is used (Demetrescu et al., 2023). Only available for horizon = 1.
...	Further arguments passed to the function which is fitting the best AR model. If ar = "auto" then the internal function auto_ar is used, if ar = "forecast" then the the function forecast::auto.arima is used. If ar is of fixed length then arima is used.

Examples

```
ivx_ar_fit(monthly$Ret, as.matrix(monthly$LTY))

ivx_ar_fit(monthly$Ret, as.matrix(monthly$LTY), ar = 1)
```

ivx_boot

*Wild Bootstrap Inference for IVX Models***Description**

Computes bootstrap p-values for the IVX Wald and t statistics using the residual wild bootstrap (RWB) or the fixed regressor wild bootstrap (FRWB) of Demetrescu et al. (2023). The null hypothesis of no predictability is imposed on the bootstrap samples. RWB rebuilds the regressor from an AR fit and its residuals (multiplied by the same wild multiplier as the predictive-regression residuals, so the innovation correlation is preserved); FRWB keeps the regressors and instruments fixed and only resamples the response.

Usage

```
ivx_boot(
  object,
  B = 999,
  type = c("rwb", "frwb"),
  ar_max = 5,
  dist = c("rademacher", "normal"),
  seed = NULL,
  cores = 1L
)

## S3 method for class 'ivx_boot'
print(x, digits = max(3L, getOption("digits") - 3L), ...)
```

Arguments

object	an object of class "ivx" (not "ivx_ar"), fitted without weights.
B	number of bootstrap replications.
type	bootstrap scheme: "rwb" (residual wild bootstrap, recommended for strongly persistent regressors) or "frwb" (fixed regressor wild bootstrap).
ar_max	maximum lag order of the (vector) autoregression fitted to the regressors by the RWB scheme; the order is selected by BIC (Remark 24).
dist	distribution of the wild multipliers.
seed	optional integer seed. With cores > 1 the L'Ecuyer-CMRG streams of the parallel package are used, so results are reproducible for a given seed and cores but differ from the serial run.
cores	number of CPU cores. Uses forking on Unix and a PSOCK cluster on Windows (the package must be installed for the workers to load it).
x	an object of class "ivx_boot".
digits	minimal number of significant digits.
...	further arguments passed to printCoefmat() .

Value

an object of class "ivx_boot": a list with the observed statistics, the bootstrap distributions (boot), and bootstrap p-values (p.value). p.value\$stat has one column per alternative: two-sided, $\beta < 0$ and $\beta > 0$.

References

Demetrescu, M., Georgiev, I., Rodrigues, P. M. M., & Taylor, A. M. R. (2023). Extensions to IVX methods of inference for return predictability. *Journal of Econometrics*, 237(2), 105271.

Examples

```
mod <- ivx(Ret ~ DP + TBL, data = kms)
ivx_boot(mod, B = 199, seed = 1)
```

Description

Tests for "pockets" of predictability using the suprema of sequences of subsample IVX statistics (Demetrescu et al., 2023, Section 3.2; Demetrescu et al., 2022). For each window the IVX statistic is computed from the window's observations with the full-sample instrument (eqs 15-17); the test statistics are the maximum (right-tailed), minimum (left-tailed) and maximum squared (two-sided) t-ratio over the sequence for a single predictor, and the maximum Wald statistic for several predictors (Remark 11). P-values are obtained by wild bootstrap (Algorithms 1-2), which the paper shows to be asymptotically valid for these sup-functionals.

Usage

```
ivx_episodic(
  object,
  scheme = c("rolling", "forward", "backward"),
  window = 0.2,
  robust = FALSE,
  B = 999,
  type = c("frwb", "rwb"),
  ar_max = 5,
  dist = c("rademacher", "normal"),
  seed = NULL,
  cores = 1L
)

## S3 method for class 'ivx_episodic'
print(x, digits = max(3L, getOption("digits") - 3L), ...)
```

Arguments

object	an object of class "ivx" fitted with horizon = 1.
scheme	"rolling" windows of fixed width, "forward" recursive windows starting at the first observation, or "backward" recursive windows ending at the last observation.
window	fraction of the sample: the window width for "rolling", the warm-in fraction τ_L for "forward", and the latest start τ_U for "backward".
robust	logical; use Eicker-White standard errors in the subsample statistics.
B	number of bootstrap replications.
type	bootstrap scheme, see ivx_boot() . The default fixed regressor wild bootstrap is the scheme used by Demetrescu et al. (2022).
ar_max	maximum lag order of the (vector) autoregression fitted to the regressors by the RWB scheme; the order is selected by BIC (Remark 24).

dist	distribution of the wild multipliers.
seed	optional integer seed. With cores > 1 the L'Ecuyer-CMRG streams of the parallel package are used, so results are reproducible for a given seed and cores but differ from the serial run.
cores	number of CPU cores. Uses forking on Unix and a PSOCK cluster on Windows (the package must be installed for the workers to load it).
x	an object of class "ivx_episodic".
digits	minimal number of significant digits.
...	unused.

Value

an object of class "ivx_episodic": the observed sequence (sequence, one row per window with its start/end index and statistics), the sup statistics (statistic) and their bootstrap p-values (p.value), plus the bootstrap draws (boot).

References

Demetrescu, M., Georgiev, I., Rodrigues, P. M. M., & Taylor, A. M. R. (2022). Testing for episodic predictability in stock returns. *Journal of Econometrics*, 227(1), 85-113.

Demetrescu, M., Georgiev, I., Rodrigues, P. M. M., & Taylor, A. M. R. (2023). Extensions to IVX methods of inference for return predictability. *Journal of Econometrics*, 237(2), 105271.

Examples

```
mod <- ivx(Ret ~ DP, data = kms)
ivx_episodic(mod, scheme = "rolling", window = 0.2, B = 99, seed = 1)
```

ivx_fit

Fitter Functions for IVX Models

Description

Basic function called by ivx to fit predictive models. These should only be used directly by experienced users.

Usage

```
ivx_fit(
  y,
  x,
  horizon = 1,
  offset = NULL,
  beta = 0.95,
  cz = 1,
  bandwidth = NULL,
```

```

    robust = FALSE,
    lag_y = FALSE,
    ...
)

ivx_wfit(
  y,
  x,
  w,
  horizon = 1,
  offset = NULL,
  beta = 0.95,
  cz = 1,
  bandwidth = NULL,
  robust = FALSE,
  lag_y = FALSE,
  ...
)

```

Arguments

y	vector of observations of length n, or a matrix with n rows.
x	design matrix of dimension n * p.
horizon	is the horizon (default horizon = 1 corresponds to a short-horizon regression). For horizon > 1 the estimator and modified Wald statistic are those of Kostakis et al. (2023), eqs (15) and (23): K-period sums of the response and of the lagged predictors, a single-lag instrument in the signal matrix and the K-period sum of the instrument in the covariance.
offset	(numeric of length n). This can be used to specify an <i>a priori</i> known component to be included in the linear predictor during fitting.
beta, cz	tuning parameters of the IVX instrument $z_t = \sum_{j=0}^{t-1} (1 - c_z/n^\beta)^j \Delta x_{t-j}$. Defaults (beta = 0.95, cz = 1) follow Kostakis et al. (2015).
bandwidth	Newey-West bandwidth for the long-run covariance estimate. The default NULL uses $n^{1/3}$ as in Kostakis et al. (2015).
robust	logical. If TRUE the Eicker-White (heteroskedasticity-robust) form of the IVX covariance matrix is used (Demetrescu et al., 2023). Only available for horizon = 1.
lag_y	logical. If TRUE the regression is augmented with the lagged dependent variable (column y_lag), instrumented by itself, as in Demetrescu (2014): this can raise the local power of the IVX test when the predictors are highly persistent and endogenous, at no cost otherwise. The joint Wald statistic still tests only the predictors. Only for horizon = 1.
...	currently disregarded.
w	vector of weights (length n) to be used in the fitting process for the wfit functions. Weighted least squares is used with weights w, i.e., $\text{sum}(w * e^2)$ is minimized.

Examples

```
ivx_fit(monthly$Ret, as.matrix(monthly$LTY))
```

ivx_iv

IV Predictability Tests with Combined Instruments

Description

`ivx_iv` implements the instrumental-variable tests of Breitung and Demetrescu (2015): the predictive regression is estimated by 2SLS with Eicker-White standard errors (their eq. 12) using instruments that are less persistent than the predictor. Two families are available. Type-I instruments are transformations of the predictor itself: the fractional difference $\Delta_+^d x_{t-1}$ ("frac") and the long difference $x_{t-1} - x_{t-1-k_T}$ ("diff"), which keep power when the predictor is stationary. Type-II instruments are deterministic and correlate with a near-integrated predictor only: the sine function $\sin(\pi t/T)$ ("sin"). Their 2SLS combination ("comb", the paper's IVcomb and the authors' recommendation) is asymptotically dominated by whichever instrument is informative, so the squared t-ratio (and the Wald statistic with several predictors) is chi-square whatever the persistence of the predictor (Theorems 3 and 7).

Usage

```
ivx_iv(
  formula,
  data,
  instruments = c("comb", "sin", "frac", "diff"),
  d = 0.5,
  kappa = 0.2,
  eta = 0.85,
  na.action,
  contrasts = NULL,
  model = TRUE,
  x = FALSE,
  y = FALSE,
  ...
)

## S3 method for class 'ivx_iv'
print(x, digits = max(3L, getOption("digits") - 3L), ...)
```

Arguments

<code>formula</code>	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
<code>data</code>	n optional data frame, list or environment (or object coercible by <code>as.data.frame</code> to a data frame) containing the variables in the model. If not found in data, the variables are taken from <code>environment(formula)</code> , typically the environment from which <code>lm</code> is called.

<code>instruments</code>	instrument set; see Details.
<code>d</code>	order of the fractional difference for "frac", in (0, 1/2]; the paper uses 1/2.
<code>kappa, eta</code>	the long-difference lag is $k_T = \lfloor \kappa T^\eta \rfloor$ (paper: 0.2 and 0.85), truncated to $t - 1$.
<code>na.action</code>	a function which indicates what should happen when the data contain NAs. The default is set by the <code>na.action</code> setting of <code>options</code> , and is <code>na.fail</code> if that is unset. The 'factory-fresh' default is <code>na.omit</code> . Another possible value is <code>NULL</code> , no action. Value <code>na.exclude</code> can be useful.
<code>contrasts</code>	an optional list. See the <code>contrasts.arg</code> of <code>model.matrix.default</code> .
<code>model</code>	logical. If TRUE the <code>model.frame</code> of the fit is returned.
<code>x</code>	an object of class "ivx_iv".
<code>y</code>	logical. If TRUE the response of the fit is returned.
<code>...</code>	additional arguments to be passed to the low level regression fitting functions (see <code>lm</code>).
<code>digits</code>	the number of significant digits to use when printing.

Details

With K predictors each type-I instrument is built per predictor and the sine instruments use frequencies $\sin(k\pi t/T)$, $k = 1, \dots, K$, so that the instrument vector stays linearly independent (Assumption 5). The IVX instrument of Kostakis et al. (2015) is the paper's "mild integration" type-I case and is available through `ivx()`.

Value

an object of class `c("ivx_iv", "ivx")` so the `ivx` methods apply; `instruments` holds the instrument matrix aligned with the regressors.

References

Breitung, J., & Demetrescu, M. (2015). Instrumental variable and variable addition based inference in predictive regressions. *Journal of Econometrics*, 187(1), 358-375.

Examples

```
ivx_iv(Ret ~ DP, data = kms)
summary(ivx_iv(Ret ~ DP + TBL, data = kms, instruments = "frac"))
```

ivx_iv_fit

*Fitter Function for IV Predictability Tests***Description**

Basic function called by `ivx_iv`. Should only be used directly by experienced users.

Usage

```
ivx_iv_fit(y, x, instruments = "comb", d = 0.5, kappa = 0.2, eta = 0.85, ...)
```

Arguments

<code>y</code>	vector of observations of length n , or a matrix with n rows.
<code>x</code>	design matrix of dimension $n \times p$.
<code>instruments</code>	instrument set; see Details.
<code>d</code>	order of the fractional difference for "frac", in $(0, 1/2]$; the paper uses $1/2$.
<code>kappa, eta</code>	the long-difference lag is $k_T = \lfloor \kappa T^\eta \rfloor$ (paper: 0.2 and 0.85), truncated to $t - 1$.
<code>...</code>	currently unused.

Examples

```
ivx_iv_fit(kms$Ret, as.matrix(kms$DP))$tstat
```

ivx_qr

*Fitting IVX Quantile Predictive Regressions***Description**

`ivx_qr` implements the IVX-QR predictability test of Lee (2016): the τ -quantile of the response is regressed on the IVX-filtered predictors (Section 3.3, Proposition 3.2), giving a test of $H_0 : \beta_\tau = 0$ with a standard chi-square limit whatever the persistence of the predictors. Requires the **quantreg** package.

Usage

```
ivx_qr(
  formula,
  data,
  tau = 0.5,
  beta = 0.95,
  cz = 5,
  na.action,
  contrasts = NULL,
```

```

    model = TRUE,
    x = FALSE,
    y = FALSE,
    ...
)

## S3 method for class 'ivx_qr'
print(x, digits = max(3L, getOption("digits") - 3L), ...)

## S3 method for class 'ivx_qr'
summary(object, ...)
```

Arguments

formula	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
data	n optional data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which lm is called.
tau	quantile level(s) in (0, 1). A vector fits one model per level.
beta, cz	tuning parameters of the IVX instrument $z_t = \sum_{j=0}^{t-1} (1 - c_z/n^\beta)^j \Delta x_{t-j}$. Defaults (beta = 0.95, cz = 1) follow Kostakis et al. (2015).
na.action	a function which indicates what should happen when the data contain NAs. The default is set by the na.action setting of options , and is na.fail if that is unset. The ‘factory-fresh’ default is na.omit . Another possible value is NULL, no action. Value na.exclude can be useful.
contrasts	an optional list. See the contrasts.arg of model.matrix.default .
model	logical. If TRUE the model.frame of the fit is returned.
x	an object of class "ivx_qr".
y	logical. If TRUE the response of the fit is returned.
...	further arguments passed to quantreg::rq() .
digits	the number of significant digits to use when printing.
object	an object of class "ivx_qr".

Details

The instrument is $z_t = (1 - c_z/n^\beta)z_{t-1} + \Delta x_t$. Lee (2016) normalises $c_z = 5$ and picks β from a look-up table indexed by the estimated QR endogeneity $\hat{\rho}(\tau) = -\text{corr}(1\{\hat{u}_t < 0\}, \hat{u}_{x,t})$, which is returned as rho_tau so the rule can be applied by the user; the default beta = 0.95 follows Kostakis et al. (2015). The sparsity $f_u(0)$ is estimated by a Gaussian kernel with Silverman’s bandwidth (footnote 4 of the paper).

Value

For a single tau, an object of class `c("ivx_qr", "ivx")` (so `summary()`, `vcov()` etc. apply) with components `tau`, `rho_tau`, `sparsity` ($\hat{f}_u(0)$) and `rq` (the underlying `quantreg::rq` fit). For several tau, a list of such objects named by tau.

References

Lee, J. H. (2016). Predictive quantile regression with persistent covariates: IVX-QR approach. *Journal of Econometrics*, 192(1), 105-118.

Examples

```
if (requireNamespace("quantreg", quietly = TRUE)) {
  summary(ivx_qr(Ret ~ DP, data = kms, tau = 0.5))
  ivx_qr(Ret ~ DP + TBL, data = kms, tau = c(0.1, 0.5, 0.9))
}
```

ivx_qr_boot

Moving Block Bootstrap for IVX-QR

Description

Percentile confidence intervals and p-values for the IVX-QR coefficients from the moving block bootstrap (MBB) of Fan and Lee (2019, Section 5). Blocks of the pairs $(y_t, \tilde{z}_{t-1})$ are resampled and the quantile regression of Lee (2016) is refitted on each bootstrap sample. This avoids estimating the sparsity and the nuisance parameters that appear under conditional heteroskedasticity, which is where the asymptotic IVX-QR test is most distorted, especially in the tails.

Usage

```
ivx_qr_boot(object, B = 999, block = NULL, level = 0.95, seed = NULL)
```

```
## S3 method for class 'ivx_qr_boot'
print(x, digits = max(3L, getOption("digits") - 3L), ...)
```

Arguments

<code>object</code>	an object of class "ivx_qr".
<code>B</code>	number of bootstrap replications.
<code>block</code>	block length; the default is $\lceil n^{1/4} \rceil$ as in the paper.
<code>level</code>	confidence level of the percentile intervals.
<code>seed</code>	optional integer seed.
<code>x</code>	an object of class "ivx_qr_boot".
<code>digits</code>	minimal number of significant digits.
<code>...</code>	unused.

Value

an object of class "ivx_qr_boot": a list with the estimates, the percentile intervals ci, two-sided percentile p-values p.value for $H_0 : \beta_j = 0$, and the bootstrap draws boot.

References

Fan, R., & Lee, J. H. (2019). Predictive quantile regressions under persistence and conditional heteroskedasticity. *Journal of Econometrics*, 213(1), 261-280.

Examples

```
if (requireNamespace("quantreg", quietly = TRUE)) {
  m <- ivx_qr(Ret ~ DP, data = kms, tau = 0.1)
  ivx_qr_boot(m, B = 199, seed = 1)
}
```

ivx_qr_fit

Fitter Function for IVX-QR Models

Description

Basic function called by ivx_qr. Should only be used directly by experienced users.

Usage

```
ivx_qr_fit(y, x, tau = 0.5, beta = 0.95, cz = 5, ...)
```

Arguments

y	vector of observations of length n, or a matrix with n rows.
x	design matrix of dimension $n \times p$.
tau	quantile level(s) in (0, 1). A vector fits one model per level.
beta, cz	tuning parameters of the IVX instrument $z_t = \sum_{j=0}^{t-1} (1 - c_z/n^\beta)^j \Delta x_{t-j}$. Defaults (beta = 0.95, cz = 1) follow Kostakis et al. (2015).
...	currently disregarded.

Examples

```
if (requireNamespace("quantreg", quietly = TRUE)) {
  ivx_qr_fit(kms$Ret, as.matrix(kms$DP), tau = 0.5)$Wald_Joint
}
```

ivx_ra

*Fitting Residual-Augmented IVX Models***Description**

`ivx_ra` implements the residual-augmented (bias-reduced) IVX estimator of Demetrescu and Rodrigues (2022). An autoregression of order p is fitted to the predictors, the predictive regression is augmented with its residuals (in the spirit of Amihud and Hurvich, 2004), and the slope on the lagged predictors is estimated by IVX. Inference uses the heteroskedasticity-robust standard errors of the paper (eq. 9 and 14), which are valid whether the predictors are stationary or near-integrated.

Usage

```
ivx_ra(
  formula,
  data,
  ar = "auto",
  ar_ic = c("aic", "bic"),
  ar_max = 5,
  horizon = 1,
  beta = 0.95,
  cz = 1,
  na.action,
  contrasts = NULL,
  model = TRUE,
  x = FALSE,
  y = FALSE,
  ...
)

## S3 method for class 'ivx_ra'
print(x, digits = max(3L, getOption("digits") - 3L), ...)
```

Arguments

<code>formula</code>	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
<code>data</code>	n optional data frame, list or environment (or object coercible by <code>as.data.frame</code> to a data frame) containing the variables in the model. If not found in data, the variables are taken from <code>environment(formula)</code> , typically the environment from which <code>lm</code> is called.
<code>ar</code>	order of the autoregression fitted to the predictors: "auto" selects it by <code>ar_ic</code> in levels (as recommended in the paper), or a positive integer for a fixed order.
<code>ar_ic</code>	information criterion for <code>ar = "auto"</code> .
<code>ar_max</code>	maximum order considered when <code>ar = "auto"</code> .

horizon	forecast horizon h ; see Details.
beta, cz	tuning parameters of the IVX instrument $z_t = \sum_{j=0}^{t-1} (1 - c_z/n^\beta)^j \Delta x_{t-j}$. Defaults (beta = 0.95, cz = 1) follow Kostakis et al. (2015).
na.action	a function which indicates what should happen when the data contain NAs. The default is set by the na.action setting of <code>options</code> , and is <code>na.fail</code> if that is unset. The ‘factory-fresh’ default is <code>na.omit</code> . Another possible value is NULL, no action. Value <code>na.exclude</code> can be useful.
contrasts	an optional list. See the <code>contrasts.arg</code> of <code>model.matrix.default</code> .
model	logical. If TRUE the <code>model.frame</code> of the fit is returned.
x	an object of class "ivx_ra".
y	logical. If TRUE the response of the fit is returned.
...	additional arguments to be passed to the low level regression fitting functions (see <code>lm</code>).
digits	the number of significant digits to use when printing.

Details

For $\text{horizon} > 1$ the estimator is the transformed-regression residual-augmented IVX of Demetrescu, Rodrigues and Taylor (2023), eqs (4.9), (4.11) and (5.5)-(5.7): the single-period response is regressed on the h -period transformed instrument $z_t^{trf,(h)} = \sum_{i=\max(1,t-h+1)}^{\min(t,T-h)} z_i$ (eq. 4.4), which accounts for the overlap of the long-horizon regression without HAC estimation. At $\text{horizon} = 1$ it coincides with the short-horizon estimator. The coefficients estimate the h -period slope β_h ; fitted values and residuals are those of the transformed (non-overlapping) regression, and the Kostakis et al. (2015) intercept correction is applied only at $\text{horizon} = 1$.

The autoregression of the predictors is fitted without an intercept and its residuals are demeaned before augmentation, which is the paper’s preferred $\tilde{t}_{ivx}^{\mu_0}$ statistic (Sections 4-5); the standard errors include the finite-sample correction of Kostakis et al. (2015) as in the paper’s simulations.

Value

an object of class `c("ivx_ra", "ivx")`; the usual `ivx` methods (`summary`, `vcov`, `delta`, ...) apply. Additional components: `gamma` (coefficients on the augmentation residuals) and `ar_order`.

References

- Demetrescu, M., & Rodrigues, P. M. M. (2022). Residual-augmented IVX predictive regression. *Journal of Econometrics*, 227(2), 429-460.
- Demetrescu, M., Rodrigues, P. M. M., & Taylor, A. M. R. (2023). Transformed regression-based long-horizon predictability tests. *Journal of Econometrics*, 237(2), 105316.
- Amihud, Y., & Hurvich, C. M. (2004). Predictive regressions: A reduced-bias estimation method. *Journal of Financial and Quantitative Analysis*, 39(4), 813-841.

Examples

```

ivx_ra(Ret ~ DP, data = kms)

summary(ivx_ra(Ret ~ DP + TBL, data = kms, ar = 2))

# long horizon (Demetrescu, Rodrigues & Taylor, 2023)
ivx_ra(Ret ~ DP, data = kms, horizon = 12)

```

ivx_ra_fit

*Fitter Function for Residual-Augmented IVX Models***Description**

Basic function called by `ivx_ra` to fit the model. Should only be used directly by experienced users.

Usage

```

ivx_ra_fit(
  y,
  x,
  ar = "auto",
  ar_ic = "aic",
  ar_max = 5,
  horizon = 1,
  beta = 0.95,
  cz = 1,
  ...
)

```

Arguments

<code>y</code>	vector of observations of length n , or a matrix with n rows.
<code>x</code>	design matrix of dimension $n \times p$.
<code>ar</code>	order of the autoregression fitted to the predictors: "auto" selects it by <code>ar_ic</code> in levels (as recommended in the paper), or a positive integer for a fixed order.
<code>ar_ic</code>	information criterion for <code>ar = "auto"</code> .
<code>ar_max</code>	maximum order considered when <code>ar = "auto"</code> .
<code>horizon</code>	forecast horizon h ; see Details.
<code>beta, cz</code>	tuning parameters of the IVX instrument $z_t = \sum_{j=0}^{t-1} (1 - c_z/n^\beta)^j \Delta x_{t-j}$. Defaults (<code>beta = 0.95</code> , <code>cz = 1</code>) follow Kostakis et al. (2015).
<code>...</code>	currently unused.

Examples

```
ivx_ra_fit(kms$Ret, as.matrix(kms$DP))
```

ivx_sys

*Fitting Systems of IVX Predictive Regressions***Description**

ivx_sys estimates a system of predictive regressions $y_t = \mu + Ax_{t-1} + \varepsilon_t$ with an m -vector response and r predictors of arbitrary persistence by IVX, and tests linear restrictions on A with the IVX-Wald statistic of Kostakis, Magdalinos and Stamatogiannis (2023), eqs (15) and (23), whose covariance has the Kronecker form $(Z'X)^{-1} \otimes I_m$ around $Z(K)'Z(K) \otimes \hat{\Sigma} - n\bar{z}\bar{z}' \otimes \hat{\Sigma}_{FM}$. Long horizons are handled as in `ivx()`. For $m = 1$ the results coincide with `ivx()`.

Usage

```
ivx_sys(
  formula,
  data,
  horizon,
  na.action,
  contrasts = NULL,
  model = TRUE,
  x = FALSE,
  y = FALSE,
  beta = 0.95,
  cz = 1,
  bandwidth = NULL,
  ...
)

## S3 method for class 'ivx_sys'
print(x, digits = max(3L, getOption("digits") - 3L), ...)

## S3 method for class 'ivx_sys'
summary(object, ...)
```

Arguments

formula	a formula whose left-hand side is a matrix, e.g. <code>cbind(y1, y2) ~ x1 + x2</code> .
data	n optional data frame, list or environment (or object coercible by <code>as.data.frame</code> to a data frame) containing the variables in the model. If not found in data, the variables are taken from <code>environment(formula)</code> , typically the environment from which <code>lm</code> is called.
horizon	is the horizon (default horizon = 1 corresponds to a short-horizon regression). For horizon > 1 the estimator and modified Wald statistic are those of Kostakis et al. (2023), eqs (15) and (23): K-period sums of the response and of the lagged predictors, a single-lag instrument in the signal matrix and the K-period sum of the instrument in the covariance.

na.action	a function which indicates what should happen when the data contain NAs. The default is set by the na.action setting of <code>options</code> , and is <code>na.fail</code> if that is unset. The ‘factory-fresh’ default is <code>na.omit</code> . Another possible value is <code>NULL</code> , no action. Value <code>na.exclude</code> can be useful.
contrasts	an optional list. See the contrasts.arg of <code>model.matrix.default</code> .
model	logical. If TRUE the model.frame of the fit is returned.
x	an object of class "ivx_sys".
y	logical. If TRUE the response of the fit is returned.
beta, cz	tuning parameters of the IVX instrument $z_t = \sum_{j=0}^{t-1} (1 - c_z/n^\beta)^j \Delta x_{t-j}$. Defaults (beta = 0.95, cz = 1) follow Kostakis et al. (2015).
bandwidth	Newey-West bandwidth for the long-run covariance estimate. The default <code>NULL</code> uses $n^{1/3}$ as in Kostakis et al. (2015).
...	additional arguments to be passed to the low level regression fitting functions (see <code>lm</code>).
digits	the number of significant digits to use when printing.
object	an object of class "ivx_sys".

Value

an object of class "ivx_sys" with the coefficient matrix `coefficients` (responses in rows, predictors in columns), matching `se`, `tstat` and `Wald_Ind` matrices, the joint Wald statistic `Wald_Joint` ($\chi^2(mr)$), the per-equation Wald statistics `Wald_Eq` ($\chi^2(r)$), and `vcov` of `vec(coefficients)` (column-major, names `response:predictor`).

References

Kostakis, A., Magdalinos, T., & Stamatogiannis, M. P. (2023). Taking stock of long-horizon predictability tests: Are factor returns predictable? *Journal of Econometrics*, 237(2), 105380.

Examples

```
ivx_sys(cbind(Ret, DE) ~ DP + TBL, data = kms)

summary(ivx_sys(cbind(Ret, DE) ~ DP + TBL, data = kms, horizon = 4))
```

ivx_sys_fit

Fitter Function for Systems IVX Models

Description

Basic function called by `ivx_sys`. Should only be used directly by experienced users.

Usage

```
ivx_sys_fit(y, x, horizon = 1, beta = 0.95, cz = 1, bandwidth = NULL, ...)
```

Arguments

y	numeric response matrix (observations in rows).
x	design matrix of dimension $n \times p$.
horizon	is the horizon (default horizon = 1 corresponds to a short-horizon regression). For horizon > 1 the estimator and modified Wald statistic are those of Kostakis et al. (2023), eqs (15) and (23): K-period sums of the response and of the lagged predictors, a single-lag instrument in the signal matrix and the K-period sum of the instrument in the covariance.
beta, cz	tuning parameters of the IVX instrument $z_t = \sum_{j=0}^{t-1} (1 - c_z/n^\beta)^j \Delta x_{t-j}$. Defaults (beta = 0.95, cz = 1) follow Kostakis et al. (2015).
bandwidth	Newey-West bandwidth for the long-run covariance estimate. The default NULL uses $n^{1/3}$ as in Kostakis et al. (2015).
...	currently disregarded.

Examples

```
ivx_sys_fit(as.matrix(kms[, c("Ret", "DE")]), as.matrix(kms[, c("DP", "TBL")]))
```

kms	<i>KMS Monthly data</i>
-----	-------------------------

Description

The dataset that was used in Kostakis et al (2015), containing a range of financial variables.

- Date: year-month-date (monthly frequency)
- DE: dividend payout ratio
- LTY: long-term yield
- DY: dividend yield
- DP: dividend-price ratio
- TBL: T-bill rate
- EP: earnings-price ratio
- BM: book-to-market value ratio
- INF: inflation rate
- DFY: default yield spread
- NTIS: net equity expansion
- TMS: term spread
- Ret: S&P 500 value-weighted log excess returns

Usage

```
kms
```

Format

A data.frame with 13 variables and 1,033 observations.

Source

<https://drive.google.com/open?id=1FdT2STH02LnIweom4AwICVf-rpVMfgV4>

kms_quarterly	<i>KMS Quarterly data</i>
---------------	---------------------------

Description

The dataset that was used in Kostakis et al (2015), containing a range of financial variables.

- Date: year-month-date
- DE: dividend payout ratio
- LTY: long-term yield
- DY: dividend yield
- DP: dividend-price ratio
- TBL: T-bill rate
- EP: earnings-price ratio
- BM: book-to-market value ratio
- INF: inflation rate
- DFY: default yield spread
- NTIS: net equity expansion
- TMS: term spread
- Ret: S&P 500 value-weighted log excess returns

Usage

```
kms_quarterly
```

Format

A data.frame with 13 variables and 345 observations.

Source

<https://drive.google.com/open?id=1FdT2STH02LnIweom4AwICVf-rpVMfgV4>

summary.ivx	Summarizing IVX Model Fits
-------------	----------------------------

Description

summary method for class "ivx".

Usage

```
## S3 method for class 'ivx'
summary(object, ...)

## S3 method for class 'summary.ivx'
print(
  x,
  digits = max(3L, getOption("digits") - 3L),
  signif.stars = getOption("show.signif.stars"),
  ...
)
```

Arguments

- object object of class "ivx", usually, a result of a call to ivx.
- ... further arguments passed to or from other methods.
- x an object of class "summary.lm", usually, a result of a call to summary.lm.
- digits the number of significant digits to use when printing.
- signif.stars logical. If TRUE, ‘significance stars’ are printed for each coefficient.

Examples

```
mod <- ivx(Ret ~ LTY, data = monthly)

summary(mod)
```

summary.ivx_ar	Summarizing IVX-AR Model Fits
----------------	-------------------------------

Description

summary method for class "ivx".

Usage

```
## S3 method for class 'ivx_ar'
summary(object, ...)

## S3 method for class 'summary.ivx_ar'
print(
  x,
  digits = max(3L, getOption("digits") - 3L),
  signif.stars = getOption("show.signif.stars"),
  ...
)
```

Arguments

object	object of class "ivx_ar", usually, a result of a call to ivx_ar.
...	further arguments passed to or from other methods.
x	an object of class "summary.lm", usually, a result of a call to summary.lm.
digits	the number of significant digits to use when printing.
signif.stars	logical. If TRUE, 'significance stars' are printed for each coefficient.

Examples

```
mod <- ivx_ar(Ret ~ LTY, data = kms)

summary(mod)
```

vcov.ivx

*Calculate Variance-Covariance Matrix for a Fitted Model Object***Description**

Calculate Variance-Covariance Matrix for a Fitted Model Object

Usage

```
## S3 method for class 'ivx'
vcov(object, complete = TRUE, ...)

## S3 method for class 'summary.ivx'
vcov(object, complete = TRUE, ...)
```

Arguments

object	a fitted ivx and summary.ivx object.
complete	logical indicating if the full variance-covariance matrix should be returned. When complete = TRUE, vcov() is compatible with coef().
...	additional arguments for method functions.

Value

A matrix of the estimated covariances between the parameter estimates of the model. This should have row and column names corresponding to the parameter names given by the `coef` method.

Examples

```
mod <- ivx(Ret ~ LTY, data = monthly)

vcov(mod)
```

ylpc	<i>YLPC Quarterly data</i>
------	----------------------------

Description

The dataset that was used in ..., containing a range of variables.

Usage

```
ylpc
```

Format

An object of class `spec_tbl_df` (inherits from `tbl_df`, `tbl`, `data.frame`) with 174 rows and 12 columns.

Source

Yang, B., Long, W., Peng, L., and Cai, Z. (2020). Autoregressive-model-based inference for predictive regressions with persistent predictors. *Journal of the American Statistical Association*, 115(530), 1029-1039. DOI: 10.1080/01621459.2019.1686392

Index

- * **datasets**
 - kms, [36](#)
 - kms_quarterly, [37](#)
 - ylpc, [40](#)
- ac_test, [3](#)
- ac_test_, [3](#)
- ac_test_bg(ac_test_), [3](#)
- ac_test_bp(ac_test_), [3](#)
- ac_test_lb(ac_test_), [3](#)
- ac_test_wald(ac_test_), [3](#)
- arm, [4](#)
- arm(), [10](#)
- arm_fit, [6](#)
- as.data.frame, [5](#), [7](#), [9](#), [11](#), [13](#), [15](#), [18](#), [25](#), [28](#), [31](#), [34](#)
- cy_test, [7](#)
- cy_test_fit(cy_test), [7](#)
- delta, [8](#)
- el_test, [9](#)
- el_test_fit(el_test), [9](#)
- elliott_cf, [10](#)
- elliott_cf_fit(elliott_cf), [10](#)
- extract.ivx, [12](#)
- extract.ivx_ar(extract.ivx), [12](#)
- hlt_test, [13](#)
- hlt_test_fit(hlt_test), [13](#)
- ivx, [15](#), [18](#), [20](#)
- ivx(), [26](#), [34](#)
- ivx_ar, [17](#)
- ivx_ar_fit, [19](#)
- ivx_boot, [20](#)
- ivx_boot(), [22](#)
- ivx_episodic, [22](#)
- ivx_fit, [23](#)
- ivx_iv, [25](#)
- ivx_iv_fit, [27](#)
- ivx_qr, [27](#)
- ivx_qr_boot, [29](#)
- ivx_qr_fit, [30](#)
- ivx_ra, [31](#)
- ivx_ra(), [10](#)
- ivx_ra_fit, [33](#)
- ivx_sys, [34](#)
- ivx_sys_fit, [35](#)
- ivx_wfit(ivx_fit), [23](#)
- kms, [36](#)
- kms_quarterly, [37](#)
- lm, [5](#), [16](#), [19](#), [26](#), [32](#), [35](#)
- model.matrix.default, [5](#), [16](#), [18](#), [26](#), [28](#), [32](#), [35](#)
- model.offset, [16](#), [18](#)
- na.exclude, [5](#), [8](#), [10](#), [11](#), [14](#), [16](#), [18](#), [26](#), [28](#), [32](#), [35](#)
- na.fail, [5](#), [8](#), [10](#), [11](#), [14](#), [16](#), [18](#), [26](#), [28](#), [32](#), [35](#)
- na.omit, [5](#), [8](#), [10](#), [11](#), [14](#), [16](#), [18](#), [26](#), [28](#), [32](#), [35](#)
- options, [5](#), [8](#), [10](#), [11](#), [14](#), [16](#), [18](#), [26](#), [28](#), [32](#), [35](#)
- print.arm(arm), [4](#)
- print.cy_test(cy_test), [7](#)
- print.el_test(el_test), [9](#)
- print.elliott_cf(elliott_cf), [10](#)
- print.hlt_test(hlt_test), [13](#)
- print.ivx(ivx), [15](#)
- print.ivx_ar(ivx_ar), [17](#)
- print.ivx_boot(ivx_boot), [20](#)
- print.ivx_episodic(ivx_episodic), [22](#)
- print.ivx_iv(ivx_iv), [25](#)
- print.ivx_qr(ivx_qr), [27](#)
- print.ivx_qr_boot(ivx_qr_boot), [29](#)
- print.ivx_ra(ivx_ra), [31](#)
- print.ivx_sys(ivx_sys), [34](#)

`print.summary.ivx(summary.ivx)`, [38](#)
`print.summary.ivx_ar(summary.ivx_ar)`,
 [38](#)
`printCoefmat()`, [21](#)

`quantreg::rq()`, [28](#)

`summary.ivx`, [38](#)
`summary.ivx_ar`, [38](#)
`summary.ivx_qr(ivx_qr)`, [27](#)
`summary.ivx_sys(ivx_sys)`, [34](#)

`vcov.ivx`, [39](#)
`vcov.summary.ivx(vcov.ivx)`, [39](#)

`ylpc`, [40](#)