Package ‘ivx’

November 24, 2020

Type Package

Title Robust Econometric Inference

Version 1.1.0


License GPL-3

URL https://github.com/kvasilopoulos/ivx

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

Depends R (>= 3.1)

Imports methods, Rcpp (>= 0.12.18)

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

Enhances texreg

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

Encoding UTF-8

Language en-US

LazyData true

RoxygenNote 7.1.1

NeedsCompilation yes

Author Kostas Vasilopoulos [cre, aut], Efthymios Pavlidis [aut]

Maintainer Kostas Vasilopoulos <k.vasilopoulo@gmail.com>

Repository CRAN

Date/Publication 2020-11-24 13:50:02 UTC
Description

Autocorrelation tests

Usage

\[ \text{ac_test}(x, \text{lag\_max} = 5) \]

Arguments

- \( x \) \hspace{1cm} \text{the residuals or an ivx object.}
- \( \text{lag\_max} \) \hspace{1cm} \text{the maximum length of lags.}

Examples

```r
obj <- ivx(hpi ~ cpi + def + int + log(res), data = ylpc)
lmtest::bgtest(hpi ~ cpi + def + int + log(res), data = ylpc)
ac_test(obj, 5)
```
Tests for autocorrelation

Description

• 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

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

Details

#' If p-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

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

```r
delta(mdl)
```

---

### delta

**Calculate the delta coefficient**

**Description**

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

**Usage**

```r
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**

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

---

### extract.ivx

**extract method for ivx objects**

**Description**

extract method for ivx objects
Usage

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

```r
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.

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

```r
ivx(
  formula,
  data,
  horizon,
  na.action,
  weights,
  contrasts = NULL,
  offset,
  model = TRUE,
  x = FALSE,
  y = FALSE,
  ...
)
```

```r
## 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**: an 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).
- **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.
- **weights**: an optional vector of weights to be used in the fitting process. Should be `NULL` or a numeric vector. If non-NULL, weighted least squares is used with weights `weights` (that is, minimizing `sum(w*e^2)`); otherwise ordinary least squares is used.
- **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", usually, a result of a call to `ivx`.
- **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.

Value

an object of class "ivx".

References


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)

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, ...
)

## 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      an 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).
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.
n.a.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.
...        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


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),
  ...
)

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).
offset (numeric of length n). This can be used to specify an a priori 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.

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 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_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, ...)

ivx_wfit(y, x, w, horizon = 1, offset = NULL, ...)

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).

offset (numeric of length n). This can be used to specify an a priori known component to be included in the linear predictor during fitting.

... 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., \( \sum(w * e^2) \) is minimized.

Examples

ivx_fit(monthly$Ret, as.matrix(monthly$LTY))
**kms**

**KMS Monthly data**

**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=1FdT2STHO2Lnlweom4AwICVf-rpVMfgV4
**kms_quarterly**  
*KMS Quarterly data*

**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=1FdT2STHO2Lnlweom4AwICVf-rpVMfgV4](https://drive.google.com/open?id=1FdT2STHO2Lnlweom4AwICVf-rpVMfgV4)
Summary

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)

Description

summary method for class "ivx".

Summarizing IVX-AR Model Fits

Description

summary method for class "ivx".
Usage

```r
## 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

```r
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

```r
## 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**

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

<table>
<thead>
<tr>
<th>ylpc</th>
<th>YLPC Quarterly data</th>
</tr>
</thead>
</table>

**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**

Index

* datasets
  kms, 11
  kms_quarterly, 12
  ylpc, 15

ac_test, 2
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
as.data.frame, 6, 8

delta, 4
extract.ivx, 4
extract.ivx_ar(extract.ivx), 4

ivx, 5, 8, 9
ivx_ar, 7
ivx_ar_fit, 9
ivx_fit, 10
ivx_wfit(ivx_fit), 10

kms, 11
kms_quarterly, 12

lm, 6, 8

model.matrix.default, 6, 8
model.offset, 6, 8

na.exclude, 6, 8
na.fail, 6, 8
na.omit, 6, 8

options, 6, 8

print.ivx(ivx), 5
print.ivx_ar(ivx_ar), 7
print.summary.ivx(summary.ivx), 13
print.summary.ivx_ar(summary.ivx_ar), 13
summary.ivx, 13
summary.ivx_ar, 13

vcov.ivx, 14
vcov.summary.ivx(vcov.ivx), 14

ylpc, 15