

Package: prais (via r-universe)

September 20, 2026

Encoding UTF-8

Type Package

Title Prais-Winsten Estimator for AR(1) Serial Correlation

Version 1.3.0.9000

Description The Prais-Winsten estimator (Prais & Winsten, 1954) takes into account AR(1) serial correlation of the errors in a linear regression model. The procedure recursively estimates the coefficients and the error autocorrelation of the specified model until sufficient convergence of the AR(1) coefficient is attained.

License GPL (>= 2)

Depends R (>= 3.6.0), sandwich, pcse

Imports stats

Suggests broom, haven, knitr, lmtest, nlme, plm, rmarkdown, testthat (>= 3.0.0), texreg, tibble

VignetteBuilder knitr

URL <https://github.com/franzmohr/prais>

BugReports <https://github.com/franzmohr/prais/issues>

Collate 'augment.prais.R' 'broom.R' 'confint.prais.R' 'data.R' 'glance.prais.R' 'plm.R' 'prais-package.R' 'prais_winsten.R' 'predict.prais.R' 'print.prais.R' 'summary.prais.R' 'print.summary.prais.R' 'pw_transform.R' 'tidy.prais.R' 'vcov.prais.R' 'vcovHC.R' 'vcovPC.R'

LazyData true

Config/testthat/edition 3

Config/roxygen2/version 8.1.0

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

Date/Publication 2026-09-20 18:22:14 UTC

RemoteUrl <https://github.com/franzmohr/prais>

RemoteRef HEAD

RemoteSha 6baa833209e13b563ab25779a0fa8e6e025cf6c0

Contents

augment.prais	2
barium	3
confint.prais	5
glance.prais	6
prais_winsten	7
predict.prais	11
summary.prais	13
tidy.prais	15
vcov.prais	17
vcovHC.prais	18
vcovPC.prais	19
Index	21

augment.prais	<i>Augment Data with Information from a Prais-Winsten Object</i>
---------------	--

Description

Adds the fitted values and residuals of a model of class "prais" to the data it was estimated from, or the predictions for new data. It is a method for the generic augment of package **broom**, which has to be installed to use it.

Usage

```
## S3 method for class 'prais'
augment(x, data = x$model, newdata = NULL, ...)
```

Arguments

x	an object of class "prais", usually, a result of a call to prais_winsten .
data	a data frame with the observations the model was estimated from. Defaults to the model frame of the object, which contains the observations in the order of argument index and without the observations that were dropped because of missing values.
newdata	an optional data frame in which to look for variables with which to predict. If it is specified, the residuals are omitted from the result, because the response is not required for a prediction.
...	not used.

Details

The fitted values and residuals are on the scale of the original data, i.e. they are those of the estimated object and not those of the Prais-Winsten transformed model, which [summary.prais](#) reports. The predictions for newdata are obtained with [predict.prais](#) and do not contain a forecast of the serially correlated part of the error term.

Value

A **tibble** that contains data, or newdata, and the columns

`.fitted` the fitted values, or the predictions for newdata.
`.resid` (only if newdata is not specified) the residuals on the scale of the original data.

See Also

[tidy.prais](#), [glance.prais](#)

Examples

```
# Generate an artificial sample
set.seed(1234567)
n <- 100
x <- sample(20:40, n, replace = TRUE)
rho <- .5

# AR(1) errors
u <- rnorm(n, 0, 5)
for (i in 2:n) {
  u[i] <- u[i] + rho * u[i - 1]
}
pw_sample <- data.frame("x" = x, "y" = 10 + 1.5 * x + u, "time" = 1:n)

# Estimate
pw <- prais_winsten(y ~ x, data = pw_sample, index = "time")

# Add fitted values and residuals
if (requireNamespace("broom", quietly = TRUE)) {
  broom::augment(pw)
}
```

barium

Barium

Description

Monthly microeconomic data from the U.S. chemical industry from February 1978 to December 1988 as used in Krupp and Pollard (1996) and subsequently re-used by Wooldridge (2000) as a textbook dataset. Raw data was obtained from <http://fmwww.bc.edu/ec-p/data/wooldridge/barium.dta>.

Usage

```
data("barium")
```

Format

A data frame with 131 rows and 31 variables:

chnimp Chinese imports, bar. chl.
bchlimp Total imports, bar. chl.
befile6 Dummy variable, which is 1 for all six months before filing.
affile6 Dummy variable, which is 1 for all six months after filing.
afdec6 Dummy variable, which is 1 for all six months after decision
befile12 Dummy variable, which is 1 for all twelve months before filing.
affile12 Dummy variable, which is 1 for all twelve months after filing.
afdec12 Dummy variable, which is 1 for all twelve months after decision.
chempi Chemical production index.
gas Gasoline production.
rtwex Exchange rate index.
spr Dummy variable, which is 1 for spring months.
sum Dummy variable, which is 1 for summer months.
fall Dummy variable, which is 1 for fall months.
lchnimp Log of chnimp.
lgas Log of gas.
lrtwex Log of rtwex.
lchempi Log of chempi.
t Time trend.
feb Dummy variable for February.
mar Dummy variable for March.
apr Dummy variable for April.
may Dummy variable for May.
jun Dummy variable for June.
jul Dummy variable for July.
aug Dummy variable for August.
sep Dummy variable for September.
oct Dummy variable for October.
nov Dummy variable for November.
dec Dummy variable for December.
percchn Percent of imports from China.

References

Krupp, C.M., & Pollard, P.S., (1996). Market responses to antidumping laws: Some evidence from the U.S. chemical industry. *Canadian Journal of Economics* 29(1), 199–227. doi:10.2307/136159
 Wooldridge, J., (2000). *Instructional Stata datasets for econometrics*. [barium]. Boston College Department of Economics.

confint.prais	<i>Confidence Intervals for the Coefficients of the Prais-Winsten Estimator</i>
---------------	---

Description

Computes confidence intervals for the coefficients of a model of class "prais".

Usage

```
## S3 method for class 'prais'
confint(object, parm, level = 0.95, ...)
```

Arguments

object	an object of class "prais", usually, a result of a call to prais_winsten .
parm	a specification of the coefficients for which the intervals are required. Either a vector of names or a vector of positions. If it is omitted, the intervals of all coefficients are computed.
level	the confidence level of the intervals. Defaults to .95.
...	not used.

Details

The intervals are based on the t distribution with the residual degrees of freedom of the model and the standard errors of [summary.prais](#), so they agree with the p-values that the summary reports. Without this method the intervals would be obtained by `confint.default`, which uses the quantiles of the normal distribution and is therefore too narrow in small samples.

Coefficients of linearly dependent variables are NA and are omitted, as in [vcov.prais](#).

Value

A matrix with a row per coefficient and the lower and upper bound of the interval in its columns, which are named after the corresponding quantiles.

See Also

[vcov.prais](#), [tidy.prais](#)

Examples

```
# Generate an artificial sample
set.seed(1234567)
n <- 100
x <- sample(20:40, n, replace = TRUE)
rho <- .5
```

```
# AR(1) errors
u <- rnorm(n, 0, 5)
for (i in 2:n) {
  u[i] <- u[i] + rho * u[i - 1]
}
pw_sample <- data.frame("x" = x, "y" = 10 + 1.5 * x + u, "time" = 1:n)

# Estimate
pw <- prais_winsten(y ~ x, data = pw_sample, index = "time")

# Confidence intervals
confint(pw)
confint(pw, parm = "x", level = .9)
```

glance.prais

*Glance at a Prais-Winsten Object***Description**

Summarises a model of class "prais" in a tidy data frame with a single row of goodness-of-fit statistics. It is a method for the generic glance of package **broom**, which has to be installed to use it.

Usage

```
## S3 method for class 'prais'
glance(x, ...)
```

Arguments

x an object of class "prais", usually, a result of a call to [prais_winsten](#).
... not used.

Details

The statistics are those of [summary.prais](#). In addition to the columns that are reported for a linear model, the result contains the AR(1) coefficient and the Durbin-Watson statistics. Both are NA if the model was estimated with `panelwise = TRUE`, because there is one coefficient per panel in that case. The full set of estimates is available as `rho` of the estimated object. Statistics that are not defined for a model, such as the F statistic of a model without regressors, are NA.

Value

A [tibble](#) with a single row and the columns

`r.squared` R^2 , the fraction of variance explained by the model.
`adj.r.squared` R^2 adjusted for the number of coefficients.

sigma	the square root of the estimated variance of the random error.
statistic	the F statistic of the model.
p.value	the p-value of the F statistic.
df	the numerator degrees of freedom of the F statistic.
df.residual	the residual degrees of freedom.
nobs	the number of observations that were used for the estimation.
rho	the final estimate of the AR(1) coefficient ρ .
dw.original	the Durbin-Watson statistic of the original linear model.
dw.transformed	the Durbin-Watson statistic of the Prais-Winsten estimator.

See Also

[tidy.prais](#), [augment.prais](#)

Examples

```
# Generate an artificial sample
set.seed(1234567)
n <- 100
x <- sample(20:40, n, replace = TRUE)
rho <- .5

# AR(1) errors
u <- rnorm(n, 0, 5)
for (i in 2:n) {
  u[i] <- u[i] + rho * u[i - 1]
}
pw_sample <- data.frame("x" = x, "y" = 10 + 1.5 * x + u, "time" = 1:n)

# Estimate
pw <- prais_winsten(y ~ x, data = pw_sample, index = "time")

# Goodness of fit
if (requireNamespace("broom", quietly = TRUE)) {
  broom::glance(pw)
}
```

prais_winsten

Prais-Winsten Estimator for AR(1) Serial Correlation

Description

The Prais-Winsten estimator takes into account AR(1) serial correlation of the errors in a linear regression model. The procedure recursively estimates the coefficients and the error autocorrelation of the specified model until sufficient convergence of the AR(1) coefficient is reached. All estimates are obtained by OLS.

Usage

```
prais_winsten(
  formula,
  data,
  index = NULL,
  max_iter = 50L,
  tol = 1e-06,
  twostep = FALSE,
  panelwise = FALSE,
  rhoweight = c("none", "T", "T1"),
  ...
)

## S3 method for class 'prais'
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. Alternatively, a fitted panel model of class "plm", from which the formula, the data and the index are taken. See 'Details'.
data	a data frame containing the variables in the model. If panel data is used, it must also contain the ID and time variables. Must not be specified if formula is a model of class "plm".
index	a character vector specifying the ID and time variables. If only one variable is provided, it is assumed to be the time variable and the data will be reordered accordingly. The specified variables must not contain NA values. Must not be specified if formula is a model of class "plm". If it is NULL, which is the default, the observations are taken in the order in which they appear in data.
max_iter	integer specifying the maximum number of allowed iterations. Default is 50.
tol	numeric specifying the maximum absolute difference between the estimator of ρ in the current and the previous iteration that has to be attained to reach convergence. Default is 1e-6.
twostep	logical. If TRUE, the estimation will stop after the first iteration.
panelwise	logical. If TRUE, ρ will be calculated for each panel separately. Default is FALSE. Only used for panel data. See 'Details'.
rhoeight	character specifying how ρ should be calculated if panelwise = TRUE. See 'Details'.
...	arguments passed to lm .
x	an object of class "prais", usually, a result of a call to prais_winsten .
digits	the number of significant digits to use when printing.

Details

If formula is a fitted panel model of class "plm", the formula, the data and the index of the panel are taken from it, and the arguments data and index must not be specified. Since the Prais-Winsten estimator obtains all its estimates by ordinary least squares, only models that were estimated with model = "pooling" are supported. The within and the random effects estimator apply a transformation of their own, which does not leave the AR(1) structure of the errors intact. Fixed effects can be estimated by adding the dummy variables to the formula, as in $y \sim x + \text{factor}(\text{id})$. The data are obtained by evaluating the data argument of the call of the model, so the object it refers to must still be available. The model frame of the fitted model cannot be used instead, because its columns are named after the terms of the formula, so that a term such as factor(id) or log(x) could not be evaluated again.

Observations with missing values in the variables of formula are dropped before estimation, as in [lm](#). The gap that a dropped observation leaves in the time variable is treated like any other gap, which is described below.

Estimates of ρ are bounded to the interval $[-1, 1]$, because the transformation of the first observation of a panel requires $(1 - \rho^2)^{(1/2)}$ to be real. If ρ attains one of those bounds, the first observation of each panel becomes zero and does not contribute to the estimates, so that the estimator effectively becomes the Cochrane-Orcutt estimator.

The time variable orders the observations and gives the distance between them. Gaps are allowed: two observations of a panel that lie k periods apart have the correlation ρ^k under an AR(1) process, so such an observation is transformed to

$$(1 + \rho^2 + \dots + \rho^{2(k-1)})^{-1/2}(x_t - \rho^k x_{t-k}),$$

which is the generalised least squares transformation evaluated at the periods that were observed. For consecutive periods this is the usual $x_t - \rho x_{t-1}$, so equally spaced data are unaffected. As a gap grows, ρ^k approaches zero and the factor approaches $(1 - \rho^2)^{(1/2)}$, so that the observation after a long gap is transformed like the first observation of a panel.

The distances are counted in steps of the smallest length that divides all differences of the time variable, so that a variable measured in, say, quarters or days is handled without further arguments, and so that the estimates do not depend on the unit the periods are expressed in. Date and POSIXct variables are supported. If the periods are not multiples of a common step, a warning is issued and the observations that surround a gap are treated as if they were consecutive. The same applies without a warning if the time variable is not numeric, such as a character vector, which only orders the observations and carries no distance between them.

The estimate of ρ itself is obtained from the residuals of an observation and its predecessor, whether or not a gap lies between them. Since the correlation across a gap is ρ^k rather than ρ , the estimate is attenuated towards zero if a large share of the observations follows a gap.

If panelwise = TRUE, twostep = FALSE and rhoweight = "none", each individual estimate of rho is re-estimated until convergence is achieved for all coefficients.

If panelwise = TRUE, the calculation of ρ can be further specified in argument rhoweight. If rhoweight = "none", ρ is assumed to be panel-specific. If rhoweight = "T", ρ is calculated as a weighted mean of panel-specific estimates, where the number of available observations per panel, i.e. T_i , is used as weight. If rhoweight = "T1", ρ is calculated as a weighted mean of panel-specific estimates, where the number of available observations per panel minus one, i.e. $T_i - 1$, is used as weight.

Value

A list of class "prais" containing the following components:

coefficients	a named vector of coefficients.
rho	the values of the AR(1) coefficient ρ from all iterations.
residuals	the residuals, that is the response minus the fitted values.
fitted.values	the fitted mean values.
rank	the numeric rank of the fitted linear model.
df.residual	the residual degrees of freedom.
call	the matched call.
terms	the terms object used.
qr	the QR decomposition of the model matrix of the Prais-Winsten transformed model, as <code>lm.fit</code> returned it. It is the decomposition the covariance matrices are obtained from, and not one of the original data.
model	the original model frame, i.e., before the Prais-Winsten transformation.
xlevels	a record of the levels of the factors used in fitting.
contrasts	the contrasts used, if the model contains factors.
index	a character specifying the ID and time variables. Only added if panel data were used.
timeid	the periods of the observations, counted in whole steps from the first period of the panel they belong to. NULL if no index was given, or if the periods are not multiples of a common step.
x	the model matrix after the Prais-Winsten transformation. Only added if the data are a single time series.
y	the response after the Prais-Winsten transformation. Only added if the data are a single time series.

Diagnostic tests

The tests of package `lmtest`, such as `dwtest`, `bgtest` and `bptest`, do not use the residuals of the model they are given. They take the model matrix and the response from the components `x` and `y` of the object and re-estimate the model by ordinary least squares. Since those components hold the transformed data, the tests describe the estimated model and not the original one. The residuals they obtain are the residuals of `summary.prais`, so that `dwtest` reports the Durbin-Watson statistic of the transformed model together with a p-value.

For panel data the components are not added, because the tests difference the residuals over all observations at once, which mixes the last observation of a panel with the first observation of the next one. The tests then fall back on the model frame, which holds the original data, and their results do not refer to the estimated model. `summary.prais` reports a Durbin-Watson statistic that respects the panels instead.

References

- Beck, N. L. and Katz, J. N. (1995): What to do (and not to do) with time-series cross-section data. *American Political Science Review* 89, 634-647.
- Prais, S. J. and Winsten, C. B. (1954): Trend Estimators and Serial Correlation. Cowles Commission Discussion Paper, 383 (Chicago).
- Wooldridge, J. M. (2013): *Introductory Econometrics. A Modern Approach*. 5th ed. Mason, OH: South-Western Cengage Learning.

Examples

```
# Generate an artificial sample
set.seed(1234567)
n <- 100
x <- sample(20:40, n, replace = TRUE)
rho <- .5

# AR(1) errors
u <- rnorm(n, 0, 5)
for (i in 2:n) {
  u[i] <- u[i] + rho * u[i - 1]
}
pw_sample <- data.frame("x" = x, "y" = 10 + 1.5 * x + u, "time" = 1:n)

# Estimate
pw <- prais_winsten(y ~ x, data = pw_sample, index = "time")
summary(pw)

# A fitted pooled panel model can be used instead of formula, data and index

if (requireNamespace("plm", quietly = TRUE)) {
  data("Grunfeld", package = "plm")
  pooled <- plm::plm(inv ~ value + capital, data = Grunfeld,
    index = c("firm", "year"), model = "pooling")
  summary(prais_winsten(pooled))
}
```

predict.prais

Predict Method for Objects of Class prais

Description

Predicted values and their standard errors based on Prais-Winsten object.

Usage

```
## S3 method for class 'prais'
predict(
  object,
  newdata = NULL,
  se.fit = FALSE,
  interval = c("none", "confidence"),
  level = 0.95,
  ...
)
```

Arguments

<code>object</code>	an object of class "prais", usually, a result of a call to prais_winsten .
<code>newdata</code>	an optional data frame in which to look for variables with which to predict. It must contain the variables that appear in formula, which do not have to be transformed beforehand. If omitted, the fitted values are used.
<code>se.fit</code>	logical. If TRUE, the standard errors of the predictions are returned as well. Defaults to FALSE.
<code>interval</code>	a character specifying the type of interval that is added to the predictions. Either "none", which is the default, or "confidence" for the confidence interval of the conditional mean.
<code>level</code>	the confidence level of the interval. Defaults to .95.
<code>...</code>	further arguments passed to or from other methods.

Details

The predictions are the conditional mean of the model, i.e. the product of the regressors and the coefficients. The AR(1) structure of the errors is not used, so the result does not contain the forecast of the serially correlated part of the error term.

The standard errors and the intervals describe that conditional mean. They are obtained from the covariance matrix of [vcov.prais](#), which comes from the Prais-Winsten transformed model, and the quantiles of the t distribution with the residual degrees of freedom of the model, so they agree with the standard errors of [summary.prais](#) and the intervals of [confint.prais](#).

Prediction intervals for an individual observation are not available, because they would require an assumption about the error of the predicted period, whose variance depends on the serial correlation and, for a genuine forecast, on the distance to the last observed period. Neither is used by the predictions, as described above.

Value

If `se.fit` is FALSE and `interval` is "none", a vector of predictions. If an interval is requested, a matrix with the columns "fit", "lwr" and "upr". If `se.fit` is TRUE, a list containing the following elements:

<code>fit</code>	a vector of predictions or, if an interval was requested, the matrix that is described above.
------------------	---

`se.fit` a vector of the standard errors of the predicted means.
`df` the residual degrees of freedom of the model.
`residual.scale` the square root of the estimated variance of the random error, as in [summary.prais](#).

References

Prais, S. J. and Winsten, C. B. (1954): Trend Estimators and Serial Correlation. Cowles Commission Discussion Paper, 383 (Chicago).

See Also

[vcov.prais](#), [confint.prais](#)

Examples

```
# Generate an artificial sample
set.seed(1234567)
n <- 100
x <- sample(20:40, n, replace = TRUE)
rho <- .5

# AR(1) errors
u <- rnorm(n, 0, 5)
for (i in 2:n) {
  u[i] <- u[i] + rho * u[i - 1]
}
pw_sample <- data.frame("x" = x, "y" = 10 + 1.5 * x + u, "time" = 1:n)

# Estimate
pw <- prais_winsten(y ~ x, data = pw_sample, index = "time")

# Predict
fcst <- predict(pw)

# Predictions with standard errors and confidence intervals
newdata <- data.frame("x" = c(25, 30, 35))
predict(pw, newdata = newdata, se.fit = TRUE)
predict(pw, newdata = newdata, interval = "confidence")
```

summary.prais

Summarising the Prais-Winsten Estimator

Description

Summary method for class "prais".

Usage

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

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

Arguments

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

Value

summary.prais returns a list of class "summary.prais", which contains the following components:

call	the matched call.
residuals	the residuals of the Prais-Winsten transformed model, i.e. the residuals the reported standard errors are based on. The residuals on the scale of the original data are available as residuals of the estimated object.
coefficients	a named vector of coefficients.
rho	the values of the AR(1) coefficient ρ from all iterations.
sigma	the square root of the estimated variance of the random error.
df	degrees of freedom, a 3-vector $(p, n-p, p^*)$, the first being the number of non-aliased coefficients, the last being the total number of coefficients.
r.squared	R^2 , the 'fraction of variance explained by the model',

$$R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2},$$

where $\bar{y}$ is the mean of y_i for $y_i = 1, \dots, N$ if there is an intercept and zero otherwise.

adj.r.squared	the above R^2 statistic ' <i>adjusted</i> ', penalising for higher p .
fstatistic	(for models including non-intercept terms) a 3-vector with the value of the F-statistic with its numerator and denominator degrees of freedom.
cov.unscaled	a $p \times p$ matrix of (unscaled) covariances of the <i>coeff[j]</i> , $j=1, \dots, p$.
dw	a named 2-vector with the Durbin-Watson statistic of the original linear model and the Prais-Winsten estimator.
terms	the terms object used.

Examples

```
# Generate an artificial sample
set.seed(1234567)
n <- 100
x <- sample(20:40, n, replace = TRUE)
rho <- .5

# AR(1) errors
u <- rnorm(n, 0, 5)
for (i in 2:n) {
  u[i] <- u[i] + rho * u[i - 1]
}
pw_sample <- data.frame("x" = x, "y" = 10 + 1.5 * x + u, "time" = 1:n)

# Estimate
pw <- prais_winsten(y ~ x, data = pw_sample, index = "time")

# The coefficient table, the AR(1) coefficient and the Durbin-Watson
# statistics of the original and of the transformed model
summary(pw)

# The standard errors are those of 'vcov'
all.equal(sqrt(diag(vcov(pw))),
          summary(pw)$coefficients[, "Std. Error"])
```

tidy.prais

Tidy a Prais-Winsten Object

Description

Summarises the coefficients of a model of class "prais" in a tidy data frame with one row per coefficient. It is a method for the generic tidy of package **broom**, which has to be installed to use it.

Usage

```
## S3 method for class 'prais'
tidy(x, conf.int = FALSE, conf.level = 0.95, ...)
```

Arguments

x	an object of class "prais", usually, a result of a call to <code>prais_winsten</code> .
conf.int	logical. If TRUE, confidence intervals are added to the result. Defaults to FALSE.
conf.level	the confidence level of the intervals. Defaults to .95.
...	not used.

Details

The estimates, standard errors, test statistics and p-values are those of [summary.prais](#). The confidence intervals are those of [confint.prais](#), which are based on the t distribution with the residual degrees of freedom of the model, so they agree with the reported p-values. Coefficients of linearly dependent variables are NA and are omitted, as in [summary.prais](#).

Value

A [tibble](#) with one row per coefficient and the columns

term	the name of the coefficient.
estimate	the estimated value of the coefficient.
std.error	the standard error of the estimate.
statistic	the t statistic of the estimate.
p.value	the two-sided p-value of the t statistic.
conf.low	(if <code>conf.int = TRUE</code>) the lower bound of the confidence interval.
conf.high	(if <code>conf.int = TRUE</code>) the upper bound of the confidence interval.

See Also

[glance.prais](#), [augment.prais](#), [confint.prais](#)

Examples

```
# Generate an artificial sample
set.seed(1234567)
n <- 100
x <- sample(20:40, n, replace = TRUE)
rho <- .5

# AR(1) errors
u <- rnorm(n, 0, 5)
for (i in 2:n) {
  u[i] <- u[i] + rho * u[i - 1]
}
pw_sample <- data.frame("x" = x, "y" = 10 + 1.5 * x + u, "time" = 1:n)

# Estimate
pw <- prais_winsten(y ~ x, data = pw_sample, index = "time")

# Tidy the coefficients
if (requireNamespace("broom", quietly = TRUE)) {
  broom::tidy(pw, conf.int = TRUE)
}
```

vcov.prais

*Covariance Matrix of the Coefficients of the Prais-Winsten Estimator***Description**

Returns the estimated covariance matrix of the coefficients of a model of class "prais".

Usage

```
## S3 method for class 'prais'
vcov(object, ...)
```

Arguments

object	an object of class "prais", usually, a result of a call to prais_winsten .
...	not used.

Details

The covariance matrix is obtained from the Prais-Winsten transformed model. It is the matrix the standard errors of [summary.prais](#) are based on and is identical to `vcovHC(object, type = "const")`. Robust alternatives are available in [vcovHC.prais](#) and [vcovPC.prais](#).

Coefficients of linearly dependent variables are NA and are omitted, as in the covariance matrix of an object of class "lm".

Value

An object of class "matrix" containing the estimate of the covariance matrix of the coefficients.

See Also

[vcovHC.prais](#), [vcovPC.prais](#), [confint.prais](#)

Examples

```
# Generate an artificial sample
set.seed(1234567)
n <- 100
x <- sample(20:40, n, replace = TRUE)
rho <- .5

# AR(1) errors
u <- rnorm(n, 0, 5)
for (i in 2:n) {
  u[i] <- u[i] + rho * u[i - 1]
}
pw_sample <- data.frame("x" = x, "y" = 10 + 1.5 * x + u, "time" = 1:n)
```

```
# Estimate
pw <- prais_winsten(y ~ x, data = pw_sample, index = "time")

# Covariance matrix and confidence intervals
vcov(pw)
confint(pw)
```

vcovHC.prais

*Semirobust Covariance Matrix Estimators***Description**

Semirobust covariance matrix estimators for models of class "prais".

Usage

```
## S3 method for class 'prais'
vcovHC(x, type = c("const", "HC1", "HC0"), ...)
```

Arguments

x	an object of class "prais", usually, the result of a call to prais_winsten .
type	a character string specifying the estimation type.
...	not used.

Details

vcovHC is a function for estimating a robust covariance matrix of parameters for the Prais-Winsten estimator. The weighting schemes specified by type are analogous to those in [vcovHC](#) in package [sandwich](#) with the caveat that only "const", "HC0" and "HC1" are available.

Value

An object of class "matrix" containing the estimate of the asymptotic covariance matrix of coefficients.

See Also

[vcovHC](#)

Examples

```
# Generate an artificial sample
set.seed(1234567)
n <- 100
x <- sample(20:40, n, replace = TRUE)
rho <- .5

# AR(1) errors
u <- rnorm(n, 0, 5)
for (i in 2:n) {
  u[i] <- u[i] + rho * u[i - 1]
}
pw_sample <- data.frame("x" = x, "y" = 10 + 1.5 * x + u, "time" = 1:n)

# Estimate
pw <- prais_winsten(y ~ x, data = pw_sample, index = "time")

# The covariance matrix of the transformed model, which 'summary' reports
vcovHC(pw, type = "const")

# Heteroskedasticity consistent alternatives
vcovHC(pw, type = "HC0")
sqrt(diag(vcovHC(pw, type = "HC1")))
```

vcovPC.prais

Extract Panel-Corrected Variance Covariance Matrix

Description

Panel-corrected covariance matrix estimators for models of class "prais".

Usage

```
## S3 method for class 'prais'
vcovPC(x, pairwise = FALSE, ...)
```

Arguments

x	an object of class "prais", usually, the result of a call to prais_winsten .
pairwise	logical. If FALSE (default), only those residuals from periods that are common to all panels are used to computed the covariances. If TRUE all observations that can be matched by period between two panels are used.
...	not used.

Details

vcovPC is a function for estimating a panel-corrected covariance matrix of parameters for the Prais-Winsten estimator.

Value

An object of class "matrix".

References

Beck, N. L. and Katz, J. N. (1995): What to do (and not to do) with time-series cross-section data. American Political Science Review 89, 634-647.

See Also

[vcovPC](#)

Examples

```
# Generate an artificial panel with AR(1) errors
set.seed(1234567)
n_id <- 8
n_time <- 25
rho <- .6

panel <- do.call(rbind, lapply(seq_len(n_id), function(i) {
  u <- rnorm(n_time, 0, 2)
  for (j in 2:n_time) {
    u[j] <- u[j] + rho * u[j - 1]
  }
  x <- rnorm(n_time, 10, 3)
  data.frame(id = i, year = seq_len(n_time), x = x, y = 5 + 2 * x + u)
}))

# Estimate
pw <- prais_winsten(y ~ x, data = panel, index = c("id", "year"))

# Panel-corrected standard errors
sqrt(diag(vcovPC(pw)))

# Match the observations of two panels by period instead of using only the
# periods that are common to all panels
sqrt(diag(vcovPC(pw, pairwise = TRUE)))
```

Index

* datasets

barium, 3

augment.prais, 2, 7, 16

barium, 3

bgtest, 10

bptest, 10

confint.prais, 5, 12, 13, 16, 17

dwtest, 10

glance.prais, 3, 6, 16

lm, 8, 9

lm.fit, 10

prais_winsten, 2, 5, 6, 7, 8, 12, 14, 15, 17–19

predict.prais, 2, 11

print.prais (prais_winsten), 7

print.summary.prais (summary.prais), 13

summary.prais, 2, 5, 6, 10, 12, 13, 13, 14, 16,
17

tibble, 3, 6, 16

tidy.prais, 3, 5, 7, 15

vcov.prais, 5, 12, 13, 17

vcovHC, 18

vcovHC.prais, 17, 18

vcovPC, 20

vcovPC.prais, 17, 19