

# Package ‘prais’

October 14, 2022

**Encoding** UTF-8

**Type** Package

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

**Version** 1.1.2

**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 (>= 3.2.0), sandwich, psc

**Imports** lmtest, stats

**RoxygenNote** 7.1.2

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

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

**Collate** 'prais\_winsten.R' 'print.prais.R' 'summary.prais.R'  
'print.summary.prais.R' 'pw\_transform.R' 'vcovHC.R' 'vcovPC.R'  
'zzz.R'

**NeedsCompilation** no

**Author** Franz X. Mohr [aut, cre]

**Maintainer** Franz X. Mohr <prais.r@outlook.com>

**Repository** CRAN

**Date/Publication** 2021-11-01 15:50:02 UTC

## R topics documented:

prais_winsten	2
summary.prais	4
vcovHC.prais	5
vcovPC.prais	6
<b>Index</b>	<b>8</b>

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,
  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.
data	a data frame containing the variables in the model. If panel data is used, it must also contain the ID and time variables.
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.
max_iter	integer specifying the maximum number of allowed iterations. Default is 50.
tol	numeric specifying the maximum absolute difference between the estimator of $\rho$ 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, $\rho$ will be calculated for each panel separately. Default is FALSE. Only used for panel data. See 'Details'.
rhoeight	character specifying how $\rho$ should be calculated if panelwise = TRUE. See 'Details'.

...	arguments passed to <code>lm</code> .
<code>x</code>	an object of class "prais", usually, a result of a call to <code>prais_winsten</code> .
<code>digits</code>	the number of significant digits to use when printing.

### Details

If  $\rho$  takes a value above 1 during the estimation process, the Prais-Winsten transformation cannot be applied to the first observations, because  $(1 - \rho^2)^{(1/2)}$  is not real. These observations are dropped during the respective iteration and the estimator effectively becomes the Cochrane-Orcutt estimator.

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  $\rho$  can be further specified in argument `rhoweight`. If `rhoweight = "none"`,  $\rho$  is assumed to be panel-specific. If `rhoweight = "T"`,  $\rho$  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"`,  $\rho$  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:

<code>coefficients</code>	a named vector of coefficients.
<code>rho</code>	the values of the AR(1) coefficient $\rho$ from all iterations.
<code>residuals</code>	the residuals, that is the response minus the fitted values.
<code>fitted.values</code>	the fitted mean values.
<code>rank</code>	the numeric rank of the fitted linear model.
<code>df.residual</code>	the residual degrees of freedom.
<code>call</code>	the matched call.
<code>terms</code>	the terms object used.
<code>model</code>	the original model frame, i.e., before the Prais-Winsten transformation.
<code>index</code>	a character specifying the ID and time variables.

### 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 Cengage.

**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)

```

---

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 <a href="#">prais_winsten</a> .
...	further arguments passed to or from other methods.
x	an object of class "summary.prais", usually, a result of a call to <a href="#">summary.prais</a> .
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, that is the response minus the fitted values.
coefficients	a named vector of coefficients.
rho	the values of the AR(1) coefficient $\rho$ 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 'adjusted', 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 $coef[j]$ , $j=1, \dots, p$ .
dw	a named 2-vector with the Durbin-Watson statistic of the original linear model and the Prais-Winsten estimator.
index	a character specifying the ID and time variables.

---

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 <a href="#">prais_winsten</a> .
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](#)

---

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 <a href="#">prais_winsten</a> .
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.

*vcovPC.prais*

7

**See Also**

[vcovPC](#)

# Index

`lm`, [3](#)

`prais_winsten`, [2](#), [3–6](#)

`print.prais (prais_winsten)`, [2](#)

`print.summary.prais (summary.prais)`, [4](#)

`summary.prais`, [4](#), [4](#)

`vcovHC`, [6](#)

`vcovHC.prais`, [5](#)

`vcovPC`, [7](#)

`vcovPC.prais`, [6](#)