Package: qarPI (via r-universe)

May 22, 2026

Type Package

Title Prediction Intervals for Quantile Autoregression

Version 0.1.0

Description Provides prediction intervals for classical (homoscedastic) autoregressive models (AR(p)) and quantile autoregressive models (QAR(p)). The package implements both percentile-based and predictive-root-based bootstrap procedures for multi-step-ahead prediction interval construction.

License GPL-3

Encoding UTF-8

LazyData true

Imports stats, quantreg

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

URL https://github.com/SilviaNovo/qarPI

BugReports https://github.com/SilviaNovo/qarPI/issues

Repository https://silvianovo.r-universe.dev

Date/Publication 2026-03-11 10:57:04 UTC

RemoteUrl https://github.com/silvianovo/qarpi

RemoteRef HEAD

RemoteSha b1eed6604f331251023b70bfa4d418695aaf8196

Contents

pi_AR_perc 2
pi_AR_proot 3
pi_BJ 5
pi_CB 6
pi_PP 8
pi_PRR 9

pi_QAR_perc 11
pi_QAR_proot 12
pi_TS 14
pi_X 15

Index **18**

`pi_AR_perc` *Prediction intervals based on AR-perc algorithm*

Description

Computes a bootstrap percentile-based prediction interval using the AR-perc algorithm for AR(p) model over a forecast horizon h.

Usage

```
pi_AR_perc(series, p = 1, h = 3, B = 1000, alpha = 0.05, tau = 0.5)
```

Arguments

`series`	Numeric vector or ts object with time series values.
`p`	Positive integer indicating the autoregressive order. Default is 1.
`h`	Positive integer indicating the prediction horizon. Default is 3.
`B`	Number of bootstrap replicates. Default is 1000.
`alpha`	Significance level. 1-`alpha` is the nominal coverage level. Default is 0.05.
`tau`	Quantile level used in estimation. Default is 0.5.

Details

This function implements the AR-perc algorithm described in Novo and Sánchez-Sellero (2025).

Value

A list with elements:

bfor Numeric matrix of bootstrap forecasts with dimension h x B.

lpi Numeric vector of lower bounds (length h).

upi Numeric vector of upper bounds (length h).

len Numeric vector of interval lengths (length h).

References

Novo, S., & Sánchez-Sellero, C. (2025). *Prediction intervals for quantile autoregression*. arXiv:2512.22018. https://arxiv.org/abs/2512.22018

Examples

```
set.seed(123)

# Simulation parameters
burn_in <- 300
n <- 25
m <- burn_in + n
coeff <- 0.6

# Simulate AR(1) process
e <- rnorm(m)
x <- numeric(m)
x[1] <- rnorm(1)

for (t in 2:m) {
  x[t] <- coeff * x[t - 1] + e[t]
}

series <- ts(x[(burn_in + 1):m])

# Compute prediction interval
out <- pi_AR_perc(series)

out$lpi
out$upi
out$len
```

---

pi_AR_proot *Prediction intervals based on AR-proot algorithm*

Description

Computes a bootstrap predictive-root-based prediction interval using the AR-proot algorithm for an AR(p) model over a forecast horizon h.

Usage

```
pi_AR_proot(series, p = 1, h = 3, B = 1000, alpha = 0.05, tau = 0.5)
```

Arguments

series	Numeric vector or ts object with time series values.
p	Positive integer indicating the autoregressive order. Default is 1.
h	Positive integer indicating the prediction horizon. Default is 3.
B	Number of bootstrap replicates. Default is 1000.
alpha	Significance level. 1-alpha is the nominal coverage level. Default is 0.05.
tau	Quantile level used in estimation. Default is 0.5.

Details

This function implements the AR-proot algorithm described in Novo and Sánchez-Sellero (2025). Predictive residuals are first obtained using a leave-one-out quantile autoregressive fit. To account for the estimation uncertainty of the autoregressive coefficients, bootstrap coefficient estimates are generated through a multiplier (random weights) bootstrap scheme. Conditional on these bootstrap coefficients, future bootstrap predictions and bootstrap observations are constructed, and predictive root replicates are defined as the difference between bootstrap observations and their corresponding bootstrap predictions. Equal-tailed prediction intervals are obtained from the empirical quantiles of the bootstrap predictive roots.

Value

A list with elements:

pfor Numeric vector of point forecasts (length h).

lpi Numeric vector of lower bounds (length h).

upi Numeric vector of upper bounds (length h).

len Numeric vector of interval lengths (length h).

References

Novo, S., & Sánchez-Sellero, C. (2025). *Prediction intervals for quantile autoregression*. arXiv:2512.22018. https://arxiv.org/abs/2512.22018

Examples

```
set.seed(123)

# Simulation parameters
burn_in <- 300
n <- 25
m <- burn_in + n
coeff <- 0.6

# Simulate AR(1) process
e <- rnorm(m)
x <- numeric(m)
x[1] <- rnorm(1)

for (t in 2:m) {
  x[t] <- coeff * x[t - 1] + e[t]
}

series <- ts(x[(burn_in + 1):m])

# Compute prediction interval
out <- pi_AR_proot(series)

out$lpi
out$upi
```

```
out$len
```

---

pi_BJ *Prediction intervals based on the Box-Jenkins method*

Description

Implements classical Box-Jenkins Gaussian prediction intervals for an AR(p) model fitted by ordinary least squares.

Usage

```
pi_BJ(series, p = 1, h = 3, alpha = 0.05)
```

Arguments

series	Numeric vector or ts object with time series values.
p	Positive integer indicating the autoregressive order. Default is 1.
h	Positive integer indicating the prediction horizon. Default is 3.
alpha	Significance level. 1-alpha is the nominal coverage level. Default is 0.05.

Details

This function implements the classical Box-Jenkins prediction intervals. The method fits an AR(p) model to the observed time series via ordinary least squares (OLS) and derives analytical prediction intervals under the assumption of Gaussian innovations.

Value

A list with elements:

pfor Numeric vector of point forecasts (length h).

lpi Numeric vector of lower bounds (length h).

upi Numeric vector of upper bounds (length h).

len Numeric vector of interval lengths (length h).

References

Box, G. E. and Jenkins, G. M. (1976). Time Series Analysis: Forecasting and Control. Holden–Day, San Francisco.

Examples

```
set.seed(123)

# Simulation parameters
burn_in <- 300
n <- 25
m <- burn_in + n
coeff <- 0.6

# Simulate AR(1) process
e <- rnorm(m)
x <- numeric(m)
x[1] <- rnorm(1)

for (t in 2:m) {
  x[t] <- coeff * x[t - 1] + e[t]
}

series <- ts(x[(burn_in + 1):m])

# Compute prediction interval
out <- pi_BJ(series,h=2)

out$lpi
out$upi
out$len
```

---

pi_CB *Prediction intervals based on the CB algorithm*

Description

Computes bootstrap percentile-based prediction intervals using the conditional bootstrap algorithm of Cao et al. (1997) for an AR(p) model over a forecast horizon h.

Usage

```
pi_CB(series, p = 1, h = 3, B = 1000, alpha = 0.05, method = c("OLS", "LAD"))
```

Arguments

series	Numeric vector or ts object with time series values.
p	Positive integer indicating the autoregressive order. Default is 1.
h	Positive integer indicating the prediction horizon. Default is 3.
B	Number of bootstrap replicates. Default is 1000.
alpha	Significance level. 1-alpha is the nominal coverage level. Default is 0.05.
method	Estimation method. One among "OLS" and "LAD". Default is "OLS".

Details

This function implements the conditional bootstrap algorithm described in Cao et al. (1997).

Value

A list with elements:

bfor Numeric matrix of bootstrap forecasts with dimension h x B.

lpi Numeric vector of lower bounds (length h).

upi Numeric vector of upper bounds (length h).

len Numeric vector of interval lengths (length h).

References

Cao, R., Febrero-Bande, M., González-Manteiga, W., Prada-Sánchez, J., and García-Jurado, I. (1997). Saving computer time in constructing consistent bootstrap prediction intervals for autoregressive processes. *Communications in Statistics-Simulation and Computation*, 26(3):961–978.

Examples

```
set.seed(123)

# Simulation parameters
burn_in <- 300
n <- 25
m <- burn_in + n
coeff <- 0.6

# Simulate AR(1) process
e <- rnorm(m)
x <- numeric(m)
x[1] <- rnorm(1)

for (t in 2:m) {
  x[t] <- coeff * x[t - 1] + e[t]
}

series <- ts(x[(burn_in + 1):m])

# Compute prediction interval
out <- pi_CB(series,h=4,alpha=0.01)

out$lpi
out$upi
out$len
```

pi_PP *Prediction intervals based on PP algorithm*

Description

Computes a bootstrap predictive-root-based prediction interval using the forward algorithm of Pan and Politis (2016) for an AR(p) model over a forecast horizon h.

Usage

```
pi_PP(series, p = 1, h = 3, B = 1000, alpha = 0.05)
```

Arguments

series	Numeric vector or ts object with time series values.
p	Positive integer indicating the autoregressive order. Default is 1.
h	Positive integer indicating the prediction horizon. Default is 3.
B	Number of bootstrap replicates. Default is 1000.
alpha	Significance level. 1-alpha is the nominal coverage level. Default is 0.05.

Details

This function implements the PP algorithm of Pan and Politis (2016), referred to in their article as Fp (forward bootstrap with predictive residuals).

Value

A list with elements:

pfor Numeric vector of point forecasts (length h).

lpi Numeric vector of lower bounds (length h).

upi Numeric vector of upper bounds (length h).

len Numeric vector of interval lengths (length h).

References

Pan, L. and Politis, D. N. (2016). Bootstrap prediction intervals for linear, nonlinear and nonparametric autoregressions. *Journal of Statistical Planning and Inference*, 177:1–27.

Examples

```
set.seed(123)

# Simulation parameters
burn_in <- 300
n <- 25
m <- burn_in + n
coeff <- 0.6

# Simulate AR(1) process
e <- rnorm(m)
x <- numeric(m)
x[1] <- rnorm(1)

for (t in 2:m) {
  x[t] <- coeff * x[t - 1] + e[t]
}

series <- ts(x[(burn_in + 1):m])

# Compute prediction interval
out <- pi_PP(series,alpha=0.10)

out$lpi
out$upi
out$len
```

---

pi_PRR *Prediction intervals based on the PRR algorithm*

Description

Computes bootstrap percentile-based prediction intervals using the forward algorithm of Pascual, Romo and Ruiz (2004) for an AR(p) model over a forecast horizon h.

Usage

```
pi_PRR(series, p = 1, h = 3, B = 1000, alpha = 0.05, method = c("LAD", "OLS"))
```

Arguments

series	Numeric vector or ts object with time series values.
p	Positive integer indicating the autoregressive order. Default is 1.
h	Positive integer indicating the prediction horizon. Default is 3.
B	Number of bootstrap replicates. Default is 1000.
alpha	Significance level. 1-alpha is the nominal coverage level. Default is 0.05.
method	Estimation method. One among "OLS" and "LAD". Default is "LAD".

Details

This function implements the forward bootstrap algorithm described in Pascual, Romo and Ruiz (2004).

Value

A list with elements:

bfor Numeric matrix of bootstrap forecasts with dimension h x B.

lpi Numeric vector of lower bounds (length h).

upi Numeric vector of upper bounds (length h).

len Numeric vector of interval lengths (length h).

References

Pascual, L., Romo, J., and Ruiz, E. (2004). Bootstrap predictive inference for ARIMA processes. *Journal of Time Series Analysis*, 25(4):449–465.

Examples

```
set.seed(123)

# Simulation parameters
burn_in <- 300
n <- 25
m <- burn_in + n
coeff <- 0.6

# Simulate AR(1) process
e <- rnorm(m)
x <- numeric(m)
x[1] <- rnorm(1)

for (t in 2:m) {
  x[t] <- coeff * x[t - 1] + e[t]
}

series <- ts(x[(burn_in + 1):m])

# Compute prediction interval
out <- pi_PRR(series,h=4)

out$lpi
out$upi
out$len
```

pi_QAR_perc *Prediction intervals based on QAR-perc algorithm*

Description

Computes bootstrap percentile-based prediction intervals using the QAR-perc algorithm for a QAR(p) model over a forecast horizon h.

Usage

```
pi_QAR_perc(series, p = 1, h = 3, B = 1000, alpha = 0.05)
```

Arguments

series	Numeric vector or ts object with time series values.
p	Positive integer indicating the autoregressive order. Default is 1.
h	Positive integer indicating the prediction horizon. Default is 3.
B	Number of bootstrap replicates. Default is 1000.
alpha	Significance level. 1-alpha is the nominal coverage level. Default is 0.05.

Details

This function implements the QAR-perc algorithm described in Novo and Sánchez-Sellero (2025).

Value

A list with elements:

bfor Numeric matrix of bootstrap forecasts with dimension h x B.

lpi Numeric vector of lower bounds (length h).

upi Numeric vector of upper bounds (length h).

len Numeric vector of interval lengths (length h).

References

Novo, S., & Sánchez-Sellero, C. (2025). *Prediction intervals for quantile autoregression*. arXiv:2512.22018. https://arxiv.org/abs/2512.22018

Examples

```
set.seed(123)

# Simulation parameters
burn_in <- 300
n <- 25
m <- burn_in + n
coeff <- 0.6
```

```
# Simulate AR(1) process
e <- rnorm(m)
x <- numeric(m)
x[1] <- rnorm(1)

for (t in 2:m) {
  x[t] <- coeff * x[t - 1] + e[t]
}

series <- ts(x[(burn_in + 1):m])

# Compute prediction interval
out <- pi_QAR_perc(series)

out$lpi
out$upi
out$len

# Simulate QAR(1) process
n <- 100
e<-runif(m)
x <- numeric(m)
x[c(1,2)] <- rt(2,3)

for (t in 3:m) {x[t]<-0.3*x[t-1]+0.7*e[t]*x[t-2]+qt(e[t],3)}
series2<-ts(x[301:m])

# Compute prediction interval

out2<-pi_QAR_perc(series2,h=4)
out2$lpi
out2$upi
out2$len
```

`pi_QAR_proot` *Prediction intervals based on QAR-proot algorithm*

Description

Computes a bootstrap predictive-root-based prediction interval using the QAR-proot algorithm for a QAR(p) model over a forecast horizon h.

Usage

```
pi_QAR_proot(series, p = 1, h = 3, B = 1000, alpha = 0.05, tau = 0.5)
```

Arguments

series	Numeric vector or ts object with time series values.
p	Positive integer indicating the autoregressive order. Default is 1.
h	Positive integer indicating the prediction horizon. Default is 3.
B	Number of bootstrap replicates. Default is 1000.
alpha	Significance level. 1-alpha is the nominal coverage level. Default is 0.05.
tau	Quantile level used in estimation. Default is 0.5.

Details

This function implements the QAR-proot algorithm described in Novo and Sánchez-Sellero (2025).

Value

A list with elements:

pfor Numeric vector of point forecasts (length h).

lpi Numeric vector of lower bounds (length h).

upi Numeric vector of upper bounds (length h).

len Numeric vector of interval lengths (length h).

References

Novo, S., & Sánchez-Sellero, C. (2025). *Prediction intervals for quantile autoregression*. arXiv:2512.22018. https://arxiv.org/abs/2512.22018

Examples

```
set.seed(123)

# Simulation parameters
burn_in <- 300
n <- 25
m <- burn_in + n
coeff <- 0.6

# Simulate AR(1) process
e <- rnorm(m)
x <- numeric(m)
x[1] <- rnorm(1)

for (t in 2:m) {
  x[t] <- coeff * x[t - 1] + e[t]
}

series <- ts(x[(burn_in + 1):m])

# Compute prediction interval
```

```
out <- pi_QAR_proot(series)

out$lpi
out$upi
out$len

#' # Simulate QAR(1) process
n <- 100
e<-runif(m)
x <- numeric(m)
x[c(1,2)] <- rt(2,3)

for (t in 3:m) {x[t]<-0.3*x[t-1]+0.7*e[t]*x[t-2]+qt(e[t],3)}
series2<-ts(x[301:m])

# Compute prediction interval

out2<-pi_QAR_proot(series2,h=1)
out2$lpi
out2$upi
out2$len
```

`pi_TS` *Prediction intervals based on the TS algorithm*

Description

Computes bootstrap percentile-based prediction intervals using the backward algorithm of Thombs and Schucany (1990) for an AR(p) model over a forecast horizon `h`.

Usage

```
pi_TS(series, p = 1, h = 3, B = 1000, alpha = 0.05)
```

Arguments

`series`	Numeric vector or ts object with time series values.
`p`	Positive integer indicating the autoregressive order. Default is 1.
`h`	Positive integer indicating the prediction horizon. Default is 3.
`B`	Number of bootstrap replicates. Default is 1000.
`alpha`	Significance level. 1-`alpha` is the nominal coverage level. Default is 0.05.

Details

This function implements the backward bootstrap algorithm proposed by Thombs and Schucany (1990).

Value

A list with elements:

bfor Numeric matrix of bootstrap forecasts with dimension h x B.

lpi Numeric vector of lower bounds (length h).

upi Numeric vector of upper bounds (length h).

len Numeric vector of interval lengths (length h).

References

Thombs, L. A. and Schucany, W. R. (1990). Bootstrap prediction intervals for autoregression. *Journal of the American Statistical Association*, 85(410):486–492.

Examples

```
set.seed(123)

# Simulation parameters
burn_in <- 300
n <- 25
m <- burn_in + n
coeff <- 0.6

# Simulate AR(1) process
e <- rnorm(m)
x <- numeric(m)
x[1] <- rnorm(1)

for (t in 2:m) {
  x[t] <- coeff * x[t - 1] + e[t]
}

series <- ts(x[(burn_in + 1):m])

# Compute prediction interval
out <- pi_TS(series,h=4)

out$lpi
out$upi
out$len
```

---

pi_X *Prediction intervals based on X algorithm*

Description

Computes a bootstrap percentile-based prediction interval using the algorithm suggested by Xiao (2012) for a QAR(p) model over a forecast horizon h.

Usage

```
pi_X(series, p = 1, h = 3, B = 1000, alpha = 0.05)
```

Arguments

series	Numeric vector or ts object with time series values.
p	Positive integer indicating the autoregressive order. Default is 1.
h	Positive integer indicating the prediction horizon. Default is 3.
B	Number of bootstrap replicates. Default is 1000.
alpha	Significance level. 1-alpha is the nominal coverage level. Default is 0.05.

Details

This function implements the X algorithm suggested by Xiao (2012).

Value

A list with elements:

bfor Numeric matrix of bootstrap forecasts with dimension h x B.

lpi Numeric vector of lower bounds (length h).

upi Numeric vector of upper bounds (length h).

len Numeric vector of interval lengths (length h).

References

Xiao, Z. (2012). Time series quantile regressions. Handbook of Statistics. Time Series Analysis: Methods and Applications, 30:213–257.

Examples

```
set.seed(123)

# Simulation parameters
burn_in <- 300
n <- 25
m <- burn_in + n
coeff <- 0.6

# Simulate AR(1) process
e <- rnorm(m)
x <- numeric(m)
x[1] <- rnorm(1)

for (t in 2:m) {
  x[t] <- coeff * x[t - 1] + e[t]
}

series <- ts(x[(burn_in + 1):m])
```

```
# Compute prediction interval
out <- pi_X(series)

out$lpi
out$upi
out$len

# Simulate QAR(1) process
n <- 100
e<-runif(m)
x <- numeric(m)
x[c(1,2)] <- rt(2,3)

for (t in 3:m) {x[t]<-0.3*x[t-1]+0.7*e[t]*x[t-2]+qt(e[t],3)}
series2<-ts(x[301:m])

# Compute prediction interval

out2<-pi_X(series2,h=4)
out2$lpi
out2$upi
out2$len
```

Index

pi_AR_perc, 2
pi_AR_proot, 3
pi_BJ, 5
pi_CB, 6
pi_PP, 8
pi_PRR, 9
pi_QAR_perc, 11
pi_QAR_proot, 12
pi_TS, 14
pi_X, 15