Package 'nowcast'

March 25, 2026

Title Economic Nowcasting with Bridge Equations and Real-Time Evaluation

Version 0.1.0

Description Provides bridge equations with optional autoregressive terms for nowcasting low-frequency macroeconomic variables (e.g. quarterly GDP) from higher-frequency indicators (e.g. monthly retail sales). Handles the ragged-edge problem where different indicators have different publication lags via mixed-frequency alignment. Includes pseudo-real-time evaluation with expanding or rolling windows, and the Diebold-Mariano test for comparing forecast accuracy following Harvey, Leybourne, and Newbold (1997) <doi:10.1016/S0169-2070(96)00719-4>. No API calls; designed to work with data from any source.

License MIT + file LICENSE

Encoding UTF-8

Language en-US

URL https://github.com/charlescoverdale/nowcast

BugReports https://github.com/charlescoverdale/nowcast/issues

RoxygenNote 7.3.3

Depends R (>= 4.1.0)

Imports cli (>= 3.6.0), grDevices, graphics, stats

Suggests spelling, testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Charles Coverdale [aut, cre, cph]

Maintainer Charles Coverdale <charlesfcoverdale@gmail.com>

Repository CRAN

Date/Publication 2026-03-25 21:00:03 UTC

Contents

`nc_aggregate` *Temporal Aggregation*

Description

Aggregate a time series from a higher frequency to a lower frequency. For example, convert monthly data to quarterly by taking the mean, sum, or last observation.

Usage

```
nc_aggregate(data, to = c("quarterly", "annual"), fun = mean)
```

Arguments

`data`	A data frame with columns `date` (Date) and `value` (numeric).
`to`	Character. Target frequency: `"quarterly"` or `"annual"`.
`fun`	Function used for aggregation (default mean).

Details

The default aggregation function is mean, which is appropriate for **flow variables** measured as rates or indices (e.g. GDP growth, CPI). For flow variables measured in levels (e.g. total retail sales), use `fun = sum`. For **stock variables** (e.g. interest rates, exchange rates), use `fun = function(x, ...) tail(x, 1)` to take the end-of-period value.

Value

A data frame with columns `date` and `value` at the target frequency.

Examples

```
monthly <- data.frame(
  date = seq(as.Date("2020-01-01"), as.Date("2020-12-01"), by = "month"),
  value = rnorm(12)
)
nc_aggregate(monthly, to = "quarterly")
nc_aggregate(monthly, to = "annual", fun = sum)
```

`nc_align` *Align Mixed-Frequency Time Series*

Description

Merges a quarterly target series with one or more higher-frequency indicator series into a single aligned dataset. Monthly indicators are aggregated to the quarterly frequency using the specified function (default: mean).

Usage

```
nc_align(target, ..., freq_ratio = 3L, agg_fun = mean)
```

Arguments

`target`	A data frame with columns `date` (Date) and `value` (numeric) containing the quarterly target variable (e.g. GDP growth).
`...`	One or more data frames, each with columns `date` and `value`. Names are used as column names in the output; if unnamed, generic names are assigned.
`freq_ratio`	Integer. The number of high-frequency observations per low-frequency period (default 3 for monthly-to-quarterly).
`agg_fun`	Function used to aggregate indicators to the target frequency (default mean).

Details

The default `agg_fun = mean` is appropriate for **flow variables** measured as rates or indices (GDP growth, CPI). For stock variables (interest rates, exchange rates), pass `agg_fun = function(x, ...) tail(x, 1)` to take end-of-period values. For flow variables in levels (total sales), use `agg_fun = sum` – but note that `sum` will understate the true quarterly total for partial quarters at the ragged edge (a warning is emitted).

When a quarter has fewer than `freq_ratio` monthly observations (a partial quarter at the ragged edge), the aggregation proceeds on the available data and a message is emitted. The `n_months` column in the output data frame records how many observations contributed to each quarterly value for each indicator.

Value

An nc_dataset object — a list with components:

data A data frame with date (quarter start dates) plus columns for the target and each indicator.

target_col Name of the target column.

indicator_cols Character vector of indicator column names.

target_freq Detected frequency of the target series.

indicator_freq Detected frequency of the indicator series.

availability A data frame summarising data availability per series.

Examples

```
# Create synthetic quarterly target and monthly indicators
target <- data.frame(
  date = as.Date(paste0(2020:2023, "-01-01")),
  value = c(0.5, -0.3, 0.8, 0.2)
)
ind1 <- data.frame(
  date = seq(as.Date("2020-01-01"), as.Date("2023-12-01"), by = "month"),
  value = rnorm(48)
)
aligned <- nc_align(target, indicator1 = ind1)
```

---

nc_available *List Available Nowcasting Methods*

Description

Returns a data frame describing the nowcasting methods implemented in the package.

Usage

```
nc_available()
```

Value

A data frame with columns method, description, and available.

Examples

```
nc_available()
```

nc_backtest *Pseudo-Real-Time Backtesting*

Description

Evaluate a nowcasting method by simulating its real-time performance on final revised data. At each evaluation step, the model is estimated using only data from periods 1 to *i*-1 (expanding window) or a rolling window ending at *i*-1, then used to produce a nowcast for period *i*. The nowcast is compared against the actual value at period *i*.

Usage

```
nc_backtest(
  formula,
  data,
  method = "bridge",
  ar_order = 1L,
  start = 10L,
  window = c("expanding", "rolling"),
  window_size = 20L,
  alpha = 0.05
)
```

Arguments

formula	A formula for the bridge equation (e.g. target ~ ind1 + ind2).
data	A data frame (or nc_dataset) with a date column.
method	Character. Currently only "bridge" is supported.
ar_order	Integer. Number of autoregressive lags of the target to include (default 1). Set to 0 for a static bridge equation.
start	Integer. Minimum number of observations before the first backtest evaluation (default 10).
window	Character. "expanding" (default) or "rolling".
window_size	Integer. Number of observations in the rolling window (ignored if window = "expanding").
alpha	Numeric. Significance level for prediction intervals (default 0.05).

Details

This is a **pseudo-real-time** exercise: it uses final revised data throughout, not the data that would have been available at each point in time (vintage data). As noted by Banbura et al. (2013, ECB WP 1564), data revisions can be material for some variables, so results may overstate true real-time accuracy.

Prediction intervals are computed using the full prediction standard error from predict.lm(..., interval = "prediction"), accounting for both residual variance and coefficient estimation uncertainty.

Value

A nowcast_backtest object with components:

results A data frame with columns date, nowcast, actual, error, ci_lower, ci_upper.

actuals The actual values used for evaluation.

method The method used.

window_type "expanding" or "rolling".

metrics A data frame with RMSE, MAE, and bias.

Examples

```
set.seed(42)
n <- 30
d <- data.frame(
  date = seq(as.Date("2015-01-01"), by = "quarter", length.out = n),
  gdp = cumsum(rnorm(n, 0.5, 0.3)),
  ind1 = cumsum(rnorm(n, 0.4, 0.2))
)
bt <- nc_backtest(gdp ~ ind1, data = d, start = 15)
bt
```

---

nc_bridge *Bridge Equation Nowcast*

Description

Estimate a bridge equation via OLS and produce a nowcast for the current or specified period. Bridge equations translate higher-frequency indicators (e.g. monthly) into estimates of a lower-frequency target (e.g. quarterly GDP). Following standard central bank practice (Baffigi et al. 2004, Runstler and Sedillot 2003), an autoregressive term for the target variable can be included via the ar_order argument.

Usage

```
nc_bridge(formula, data, newdata = NULL, ar_order = 1L, alpha = 0.05)
```

Arguments

formula	A formula with the target variable on the left-hand side and indicator variables on the right (e.g. target ~ ind1 + ind2).
data	A data frame (or nc_dataset) containing the variables in formula plus a date column.
newdata	A one-row data frame with indicator values for the nowcast period. If NULL, the last complete row of data is used.
ar_order	Integer. Number of autoregressive lags of the target to include (default 1). Set to 0 for a static bridge equation with no AR terms.
alpha	Numeric. Significance level for prediction intervals (default 0.05).

Details

Prediction intervals are computed using the full prediction standard error, which accounts for both residual variance and estimation uncertainty in the coefficients, evaluated against a t distribution with appropriate degrees of freedom.

When `ar_order > 0`, the model includes a lagged dependent variable. The reported standard errors assume homoskedastic, serially uncorrelated errors. If residuals exhibit autocorrelation (indicated by the Durbin-Watson statistic in `details$dw_stat`), consider extracting the fitted model via `result$model` and applying HAC standard errors from the **sandwich** package.

Value

A `nowcast_result` object.

References

Baffigi, A., Golinelli, R. and Parigi, G. (2004). Bridge models to forecast the euro area GDP. *International Journal of Forecasting*, 20(3), 447–460.

Examples

```
# Synthetic example
set.seed(42)
d <- data.frame(
  date = as.Date(paste0(2015:2024, "-01-01")),
  gdp = cumsum(rnorm(10, 0.5, 0.3)),
  ind1 = cumsum(rnorm(10, 0.4, 0.2)),
  ind2 = cumsum(rnorm(10, 0.3, 0.4))
)
nc_bridge(gdp ~ ind1 + ind2, data = d)
```

`nc_compute` *Compute a Nowcast by Method Name*

Description

A generic dispatcher that calls the appropriate `nc_*` function based on a string method name. Useful for programmatic workflows.

Usage

```
nc_compute(data, method = "bridge", ...)
```

Arguments

`data`	A data frame or `nc_dataset`.
`method`	Character. Name of the method: `"bridge"`.
`...`	Additional arguments passed to the underlying function.

Value

A nowcast_result object.

Examples

```
set.seed(42)
d <- data.frame(
  date = as.Date(paste0(2015:2024, "-01-01")),
  gdp = cumsum(rnorm(10, 0.5, 0.3)),
  ind1 = cumsum(rnorm(10, 0.4, 0.2))
)
nc_compute(d, method = "bridge", formula = gdp ~ ind1)
```

---

nc_dm_test *Diebold-Mariano Test for Equal Predictive Accuracy*

Description

Tests whether two sets of forecast errors have equal predictive accuracy. Implements the modified test of Harvey, Leybourne, and Newbold (1997), which applies a finite-sample correction to the original Diebold and Mariano (1995) statistic and uses the *t* distribution rather than the normal. The Bartlett (triangular) kernel is used for HAC variance estimation, which guarantees a non-negative variance estimate.

Usage

```
nc_dm_test(
  e1,
  e2,
  alternative = c("two.sided", "less", "greater"),
  h = 1L,
  loss = c("squared", "absolute")
)
```

Arguments

e1	Numeric vector. Forecast errors from model 1.
e2	Numeric vector. Forecast errors from model 2 (same length).
alternative	Character. "two.sided", "less" (model 1 better), or "greater" (model 2 better).
h	Integer. Forecast horizon (default 1). Used for the Newey-West bandwidth and HLN correction.
loss	Character. Loss function: "squared" or "absolute".

Value

A list with components `statistic`, `p_value`, `alternative`, `method`, and `n`.

References

Diebold, F.X. and Mariano, R.S. (1995). Comparing predictive accuracy. *Journal of Business & Economic Statistics*, 13(3), 253–263. doi:10.1080/07350015.1995.10524599

Harvey, D., Leybourne, S. and Newbold, P. (1997). Testing the equality of prediction mean squared errors. *International Journal of Forecasting*, 13(2), 281–291. doi:10.1016/S01692070(96)007194

Examples

```
set.seed(1)
e1 <- rnorm(50, sd = 1)
e2 <- rnorm(50, sd = 1.5)
nc_dm_test(e1, e2)
```

`nc_evaluate` *Evaluate Nowcast Accuracy*

Description

Compute standard forecast evaluation metrics: root mean squared error (RMSE), mean absolute error (MAE), and mean bias.

Usage

```
nc_evaluate(forecast, actual)
```

Arguments

`forecast`	Numeric vector of nowcast/forecast values.
`actual`	Numeric vector of realised values (same length as `forecast`).

Value

A data frame with one row and columns `rmse`, `mae`, and `bias`.

Examples

```
nc_evaluate(c(1.0, 2.0, 3.0), c(1.1, 1.8, 3.2))
```

nc_ragged_edge *Summarise Ragged-Edge Data Availability*

Description

Shows which series have data through which date, highlighting the ragged edge where indicators have different publication lags.

Usage

```
nc_ragged_edge(data)
```

Arguments

data	An nc_dataset object, or a data frame with a date column.

Value

A data frame with columns series, first_date, last_date, n_obs, and n_missing.

Examples

```
target <- data.frame(
  date = as.Date(c("2020-01-01", "2020-04-01", "2020-07-01")),
  value = c(0.5, -0.3, 0.8)
)
ind <- data.frame(
  date = seq(as.Date("2020-01-01"), as.Date("2020-09-01"), by = "month"),
  value = rnorm(9)
)
ds <- nc_align(target, indicator = ind)
nc_ragged_edge(ds)
```

nc_transform *Stationarity Transformations*

Description

Apply common transformations to make a time series stationary. Supports first differencing, log differencing (growth rates), and standardisation.

Usage

```
nc_transform(data, method = c("diff", "log_diff", "standardize"))
```

Arguments

data	A data frame with columns date and value, or a numeric vector.
method	Character. One of "diff" (first difference), "log_diff" (log first difference, i.e. percentage growth rate), or "standardize" (zero mean, unit variance).

Value

A data frame with columns date and value (if input was a data frame), or a numeric vector (if input was numeric). The output is shorter by one observation for "diff" and "log_diff".

Examples

```
df <- data.frame(
  date = seq(as.Date("2020-01-01"), as.Date("2020-06-01"), by = "month"),
  value = c(100, 102, 101, 105, 103, 108)
)
nc_transform(df, method = "diff")
nc_transform(df, method = "log_diff")
nc_transform(df, method = "standardize")
```

---

plot.nowcast_backtest *Plot Method for Backtest Results*

Description

Plots nowcast versus actual values over the backtest evaluation period.

Usage

```
## S3 method for class 'nowcast_backtest'
plot(x, ...)
```

Arguments

x	A nowcast_backtest object.
...	Additional arguments passed to plot().

Value

The input object, invisibly.

plot.nowcast_result *Plot Method for Nowcast Results*

Description

Plots actual versus fitted values from the nowcast model, with the nowcast point highlighted.

Usage

```
## S3 method for class 'nowcast_result'
plot(x, ...)
```

Arguments

x	A nowcast_result object.
...	Additional arguments passed to plot().

Value

The input object, invisibly.

predict.nowcast_result *Predict Method for Nowcast Results*

Description

Generate predictions from a fitted nowcast model for new indicator data.

Usage

```
## S3 method for class 'nowcast_result'
predict(object, newdata, ...)
```

Arguments

object	A nowcast_result object.
newdata	A data frame with indicator values.
...	Additional arguments (currently unused).

Value

A numeric vector of predictions.

print.nc_dataset *Print Method for nc_dataset Objects*

Description

Print Method for nc_dataset Objects

Usage

```
## S3 method for class 'nc_dataset'
print(x, ...)
```

Arguments

x	An nc_dataset object.
...	Additional arguments (currently unused).

Value

The input object, invisibly.

print.nowcast_backtest *Print Method for Backtest Results*

Description

Print Method for Backtest Results

Usage

```
## S3 method for class 'nowcast_backtest'
print(x, ...)
```

Arguments

x	A nowcast_backtest object.
...	Additional arguments (currently unused).

Value

The input object, invisibly.

print.nowcast_result *Print Method for Nowcast Results*

Description

Print Method for Nowcast Results

Usage

```
## S3 method for class 'nowcast_result'
print(x, ...)
```

Arguments

x	A nowcast_result object.
...	Additional arguments (currently unused).

Value

The input object, invisibly.

summary.nowcast_backtest *Summary Method for Backtest Results*

Description

Summary Method for Backtest Results

Usage

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

Arguments

object	A nowcast_backtest object.
...	Additional arguments (currently unused).

Value

The input object, invisibly.

summary.nowcast_result *Summary Method for Nowcast Results*

Description

Summary Method for Nowcast Results

Usage

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

Arguments

object	A nowcast_result object.
...	Additional arguments (currently unused).

Value

The input object, invisibly.

Index