Package: wickra (via r-universe)

November 3, 2026

Type Package

Title Streaming-First Technical Indicators

Version 2.1.1

Description R bindings for the Wickra technical-analysis library over its C ABI hub. Exposes 515 indicators, each an incremental streaming state machine shared with the Rust core and the other language bindings, so that live and historical evaluation use the exact same implementation.

License MIT + file LICENSE | Apache License 2.0

URL https://github.com/wickra-lib/wickra, https://docs.wickra.org, https://wickra-lib.r-universe.dev

BugReports https://github.com/wickra-lib/wickra/issues

X-schema.org-keywords technical-analysis, indicators, streaming, trading, finance, quant, algorithmic-trading, backtesting, time-series, ta-lib, candlestick, market-data

Encoding UTF-8

NeedsCompilation yes

SystemRequirements the Wickra C ABI shared library, downloaded automatically at install time from the matching GitHub release and bundled into the package. Set WICKRA_INCLUDE_DIR and WICKRA_LIB_DIR to build against a locally built C ABI instead (e.g. after `cargo build -p wickra-c --release`).

Roxygen list(markdown = TRUE)

Depends R (>= 4.1)

Suggests testthat (>= 3.0.0), knitr, rmarkdown

VignetteBuilder knitr

LazyData true

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

Config/pak/sysreqs libclang-dev

Repository https://wickra-lib.r-universe.dev

Date/Publication 2026-10-28 04:21:02 UTC

RemoteUrl https://github.com/wickra-lib/wickra

RemoteRef v2.1.1

RemoteSha a8bccd2bfa458648eecc4a54af413e9f38b6897b

RemoteSubdir bindings/r

Contents

batch 2
batch_fast 3
flush.wickra_indicator 4
is_ready 4
name 5
push 6
read 6
reset 7
sample_ohlcv 8
update.wickra_indicator 8
warmup_period 9

Index **10**

`batch` *Run an indicator over a whole series in one call*

Description

The result is identical to feeding the same inputs through `update()` one at a time, with `NA` at warmup positions.

Usage

```
batch(object, ...)

## S3 method for class 'wickra_indicator'
batch(object, ...)
```

Arguments

`object`	A `wickra_indicator`.
`...`	The input columns, and for the snapshot families the per-bar width as a named argument.

Details

Most indicators take one column per input field, all the same length. The cross-section and order-book families take a per-bar snapshot instead: each field arrives as one flat column with bar `i` at `[i*width, (i+1)*width)`, and the width is passed as a named argument in the position the native routine declares it (`members` for a cross-section, `n_bids` / `n_asks` for an order book). Naming it is what distinguishes a width from a column, so the rule that keeps a short column from being read past its end still applies to the rest.

Value

A numeric vector for a scalar indicator; an `n x k` matrix for a multi-output or profile indicator, with the field names as columns; and for a bar builder the bars the series completed, which is as many as the data makes it rather than one per input row.

Examples

```
batch(Sma(3), c(1, 2, 3, 4, 5)) # NA NA 2 3 4
```

---

batch_fast *Run an indicator over a whole series with its fast kernel*

Description

The opt-in fast form of `batch()`. Where the indicator has a SIMD kernel (moving averages, RSI, ATR, MACD, Bollinger, Chaikin, skewness, Pearson and more), the kernel reassociates the arithmetic, so each value agrees with `batch()` to within a few units in the last place rather than bit for bit; `NA` placement and the shape are identical, and the result is the same on every platform. Where there is no kernel it is exactly `batch()`.

Usage

```
batch_fast(object, ...)

## S3 method for class 'wickra_indicator'
batch_fast(object, ...)
```

Arguments

`object`	A `wickra_indicator`.
`...`	The input columns, and for the snapshot families the per-bar width as a named argument.

Value

The same shape as `batch()`.

Examples

```
batch_fast(Ema(3), c(1, 2, 3, 4, 5))
```

flush.wickra_indicator *Flush a resampler's final candle*

Description

Emit the final, still-open candle a `Resampler()` is aggregating (the partial higher-timeframe bar that no later input has closed yet). Extends the base generic `base::flush()`.

Usage

```
## S3 method for class 'wickra_indicator'
flush(con)
```

Arguments

con	A `wickra_indicator` created by `Resampler()`.

Value

A named numeric vector (`open`, `high`, `low`, `close`, `volume`, `timestamp`), or `NULL` if nothing is pending.

Examples

```
r <- Resampler(5)
for (t in 0:6) push(r, 100 + t, 101 + t, 99 + t, 100 + t, 10, t)
flush(r)
```

is_ready *Whether an indicator has consumed enough input to emit a value*

Description

Not available for the alt-chart bar builders, which have no warmup.

Usage

```
is_ready(object)

## S3 method for class 'wickra_indicator'
is_ready(object)
```

Arguments

object	A `wickra_indicator`.

Value

A single logical.

Examples

```
sma <- Sma(3)
is_ready(sma) # FALSE
for (x in c(1, 2, 3)) update(sma, x)
is_ready(sma) # TRUE
```

---

name *Canonical name of an indicator*

Description

Returns the stable, human-readable name of the indicator (the same name reported by every other Wickra binding), e.g. `"SMA"` for `Sma()`.

Usage

```
name(object)

## S3 method for class 'wickra_indicator'
name(object)
```

Arguments

`object`	A `wickra_indicator`.

Value

A single character string.

Examples

```
name(Sma(14)) # "SMA"
```

push *Push one input into a streaming aggregator*

Description

Feeds a `TickAggregator()` one trade tick (`price`, `size`, `timestamp`) or a `Resampler()` one candle (`open`, `high`, `low`, `close`, `volume`, `timestamp`), and returns the candles it closed as a numeric matrix with columns `open`, `high`, `low`, `close`, `volume`, `timestamp`. The matrix has zero rows while the open bar merely grows, and more than one row when gap filling emits placeholders for skipped buckets.

Usage

```
push(object, ...)

## S3 method for class 'wickra_indicator'
push(object, ...)
```

Arguments

`object`	A `wickra_indicator` created by `TickAggregator()` or `Resampler()`.
`...`	The input fields, in the order that object takes them: a tick is `price, size, timestamp`; a candle is `open, high, low, close, volume, timestamp`.

Value

A numeric matrix with six named columns (possibly zero rows).

Examples

```
agg <- TickAggregator(1000, FALSE)
push(agg, 100, 1, 0)
push(agg, 102, 1, 1000) # closes the first bucket

r <- Resampler(5, TRUE)
push(r, 100, 101, 99, 100, 10, 0)
```

read *Read every candle parsed by a CSV candle reader*

Description

Returns all the candles a `CandleReader()` parsed from its CSV, as a numeric matrix with columns `open`, `high`, `low`, `close`, `volume`, `timestamp`.

Usage

```
read(object)

## S3 method for class 'wickra_indicator'
read(object)
```

Arguments

object	A `wickra_indicator` created by `CandleReader()`.

Value

A numeric matrix with six named columns (zero rows for an empty CSV).

Examples

```
r <- CandleReader("timestamp,open,high,low,close,volume\n0,100,101,99,100.5,10\n")
read(r)
```

---

reset *Reset an indicator to its warmup state*

Description

Reset an indicator to its warmup state

Usage

```
reset(object)

## S3 method for class 'wickra_indicator'
reset(object)
```

Arguments

object	A `wickra_indicator`.

Value

The indicator, invisibly.

Examples

```
sma <- Sma(3)
update(sma, 1)
reset(sma)
```

sample_ohlcv *Synthetic daily OHLCV sample series*

Description

A deterministic, synthetic daily OHLCV (open / high / low / close / volume) price series for use in the examples, the *Getting started* vignette, and tests. It is a seeded random walk, **not** real market data. Regenerate with `data-raw/sample_ohlcv.R`.

Usage

```
sample_ohlcv
```

Format

A data frame with 250 rows and 6 columns:

date Trading date (`Date`).

open Opening price.

high Session high.

low Session low.

close Closing price.

volume Traded volume.

update.wickra_indicator *Update an indicator with one observation*

Description

Update an indicator with one observation

Usage

```
## S3 method for class 'wickra_indicator'
update(object, ...)
```

Arguments

`object`	A `wickra_indicator` created by an indicator constructor.
`...`	The observation: a single value for scalar indicators, the OHLCV fields plus a timestamp for candle indicators, or two values for pairwise indicators.

Value

The indicator value: a numeric scalar; a named numeric vector for multi-output indicators (NA during warmup); a matrix of completed bars for bar builders; or a list / numeric vector for profile indicators (NULL during warmup).

Examples

```
sma <- Sma(3)
for (x in c(1, 2, 3, 4, 5)) v <- update(sma, x)
v # 4
```

`warmup_period` *Number of updates an indicator needs before it produces a value*

Description

Not available for the alt-chart bar builders (RenkoBars(), KagiBars(), PointAndFigureBars(), ...), which have no warmup.

Usage

```
warmup_period(object)

## S3 method for class 'wickra_indicator'
warmup_period(object)
```

Arguments

object	A wickra_indicator.

Value

A single integer: the warmup period.

Examples

```
warmup_period(Sma(14)) # 14
```

Index

∗ **datasets**
sample_ohlcv, 8

base::flush(), *4*
batch, 2
batch(), *3*
batch_fast, 3

CandleReader(), *6, 7*

flush.wickra_indicator, 4

is_ready, 4

KagiBars(), *9*

name, 5

PointAndFigureBars(), *9*
push, 6

read, 6
RenkoBars(), *9*
Resampler(), *4*, *6*
reset, 7

sample_ohlcv, 8
Sma(), *5*

TickAggregator(), *6*

update(), *2*
update.wickra_indicator, 8

warmup_period, 9