Package: klineR (via r-universe)

October 8, 2026

Title Candlestick Pattern Detection and Stock Screening

Version 0.1.1

Description Detects classical candlestick patterns and structure-based chart patterns from OHLCV time series and provides reusable stock-screening workflows. Built-in detectors include single- and multi-candle patterns, trend structures such as double bottoms and ascending triangles, and a configurable ``golden pit'' recovery setup. Includes a unified API to run pattern scans across one or many symbols.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Imports stats

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

URL https://github.com/statlearner123/klineR

BugReports https://github.com/statlearner123/klineR/issues

Repository https://statlearner123.r-universe.dev

Date/Publication 2026-05-17 04:43:29 UTC

RemoteUrl https://github.com/statlearner123/kliner

RemoteRef HEAD

RemoteSha e52ed5bd74ff030424a6dbf7b89c7285ef349cec

Contents

available_patterns 2
demo_ohlcv 2
detect_pattern 3
pattern_catalog 4

plot_kline_patterns 4
run_simulation_demo 5
scan_patterns 6
screen_symbols 6
simulate_symbols 7

Index **8**

`available_patterns` *Built-in Candlestick Pattern Names*

Description

Returns supported pattern identifiers, Chinese names, or both.

Usage

```
available_patterns(language = c("id", "cn", "both"))
```

Arguments

`language`	Output language: `"id"`, `"cn"`, or `"both"`.

Value

A character vector of supported pattern names.

Examples

```
available_patterns()
available_patterns("cn")
```

`demo_ohlcv` *Example OHLCV Data*

Description

Produces reproducible synthetic OHLCV data for examples and tests.

Usage

```
demo_ohlcv(n = 120, seed = 123)
```

Arguments

`n`	Number of rows to generate.
`seed`	Random seed.

Value

A data frame with `date`, `open`, `high`, `low`, `close`, `volume`.

Examples

```
d <- demo_ohlcv(60, seed = 99)
head(d, 3)
```

`detect_pattern` *Detect a Candlestick or Structure Pattern*

Description

Detects one built-in pattern from OHLCV data.

Usage

```
detect_pattern(data, pattern, ...)
```

Arguments

`data`	A data frame with at least `open`, `high`, `low`, `close` columns. Optional columns include `date` and `volume`.
`pattern`	One value from `available_patterns()`.
`...`	Additional parameters passed to the selected detector.

Value

A logical vector of length `nrow(data)`. `TRUE` marks rows where the pattern is detected.

Examples

```
d <- demo_ohlcv(120)
hit <- detect_pattern(d, "double_bottom")
tail(hit, 3)
```

pattern_catalog *Official TongHuaShun Pattern Catalog*

Description

Returns the full 68-pattern catalog grouped by the official menu categories.

Usage

```
pattern_catalog()
```

Value

A data frame with `pattern_id`, `pattern_cn`, and `category`.

Examples

```
tbl <- pattern_catalog()
nrow(tbl)
```

plot_kline_patterns *Plot K-line with Pattern Annotations*

Description

Draws candlesticks with optional pattern markers for visual verification.

Usage

```
plot_kline_patterns(
  data,
  pattern = NULL,
  detections = NULL,
  main = NULL,
  max_labels = 10,
  file = NULL
)
```

Arguments

`data`	OHLCV data frame.
`pattern`	Optional pattern identifier or Chinese name.
`detections`	Optional logical vector from `detect_pattern()`. If `NULL`, detections are computed from `pattern`.
`main`	Plot title.
`max_labels`	Maximum number of labels to draw.
`file`	Optional PNG file path. If supplied, the plot is saved to disk.

Value

Invisibly returns detection indices.

Examples

```
d <- demo_ohlcv(140, seed = 1)
plot_kline_patterns(d, pattern = "doji", max_labels = 5)
```

`run_simulation_demo` *Run a Simulated Screening Demo with Charts*

Description

Simulates several symbols, runs pattern screening, and writes chart images with pattern labels.

Usage

```
run_simulation_demo(
  symbols = c("AAA", "BBB", "CCC"),
  patterns = c("double_bottom", "golden_pit", "ascending_triangle"),
  n = 220,
  seed = 123,
  output_dir = tempdir()
)
```

Arguments

`symbols`	Character vector of symbols.
`patterns`	Patterns to screen.
`n`	Number of rows per symbol.
`seed`	Random seed.
`output_dir`	Output directory for PNG charts.

Value

A list with `data`, `screen`, and `plot_files`.

Examples

```
out <- run_simulation_demo(
  symbols = c("AAA", "BBB", "CCC"),
  patterns = c("double_bottom", "golden_pit"),
  n = 180
)
out$screen
```

scan_patterns *Scan Multiple Patterns on a Single Symbol*

Description

Runs several detectors and reports whether each one is present on the latest candle.

Usage

```
scan_patterns(data, patterns = available_patterns(), ...)
```

Arguments

data	A data frame with OHLCV columns.
patterns	Character vector of pattern names.
...	Additional parameters passed to each detector.

Value

A data frame with columns pattern, detected_latest.

Examples

```
d <- demo_ohlcv(120)
scan_patterns(d, patterns = c("doji", "hammer", "golden_pit"))
```

screen_symbols *Screen Symbols by a Pattern*

Description

Applies one detector to many symbols.

Usage

```
screen_symbols(data_list, pattern, latest_only = TRUE, ...)
```

Arguments

data_list	Named list of OHLCV data frames.
pattern	Pattern identifier from available_patterns().
latest_only	If TRUE, keep symbols where the latest row is detected. If FALSE, keep symbols with at least one detection in their series.
...	Additional parameters passed to detect_pattern().

Value

A data frame with detected symbols and context columns.

Examples

```
d1 <- demo_ohlcv(130, seed = 1)
d2 <- demo_ohlcv(130, seed = 2)
screen_symbols(
  list(SYMA = d1, SYMB = d2),
  pattern = "doji",
  latest_only = FALSE
)
```

---

simulate_symbols *Simulate Multiple Symbol OHLCV Series*

Description

Generates synthetic OHLCV data for several symbols and injects representative chart structures so pattern-screening workflows can be tested end-to-end.

Usage

```
simulate_symbols(symbols = c("AAA", "BBB", "CCC"), n = 220, seed = 123)
```

Arguments

symbols	Character vector of symbol names.
n	Number of rows per symbol.
seed	Random seed.

Value

A named list of OHLCV data frames.

Examples

```
sim <- simulate_symbols(c("AAA", "BBB"), n = 180, seed = 7)
names(sim)
```

Index

available_patterns, 2
available_patterns(), *3*, *6*

demo_ohlcv, 2
detect_pattern, 3
detect_pattern(), *4*, *6*

pattern_catalog, 4
plot_kline_patterns, 4

run_simulation_demo, 5

scan_patterns, 6
screen_symbols, 6
simulate_symbols, 7