# Package: stlpro (via r-universe)

November 7, 2026

**Type** Package

**Title** STL with Extended Capabilities

**Version** 0.1.0

**Description** Implements a Seasonal-Trend decomposition procedure based on LOESS (STL, developed by Cleveland, Cleveland, McRae, and Terpenning, 1990) in 'R' and 'C++'. The package supports missing values, fractional seasonal periods, multiplicative decompositions, and cross-validation based selection of the seasonal smoothing window.

**License** GPL (>= 3)

**Encoding** UTF-8

**Imports** Rcpp

**LinkingTo** Rcpp

**Suggests** knitr, rmarkdown, testthat (>= 3.0.0)

**Config/testthat/edition** 3

**VignetteBuilder** knitr

**Config/roxygen2/version** 8.1.1

**NeedsCompilation** yes

**Author** Daniel Ollech [aut, cre], Pavithra Purushothaman [ctb]

**Maintainer** Daniel Ollech <daniel.ollech@bundesbank.de>

**Repository** https://cran.r-universe.dev

**Date/Publication** 2026-11-07 16:57:57 UTC

**RemoteUrl** https://github.com/cran/stlpro

**RemoteRef** HEAD

**RemoteSha** c86f01224cfbec5d7251a14b00e9855f4017a355

## Contents

plot.stlpro . . . . . 2
print.cv_results . . . . . 3
stl . . . . . 3

**Index** **6**

`plot.stlpro` *Plot an STL decomposition fitted by stlpro Plot the reconstructed observed series and its seasonal, trend, and remainder components.*

## Description

Plot an STL decomposition fitted by stlpro

Plot the reconstructed observed series and its seasonal, trend, and remainder components.

## Usage

```
## S3 method for class 'stlpro'
plot(
  x,
  plot_type = c("multiple", "single"),
  multiplicative = NULL,
  xlab = "",
  ylab = "data",
  main = NULL,
 labels = c("Unadjusted and adjusted series", "Seasonal component", "Trend component",
    "Remainder component"),
  col = c("#071108", "#D31B0D", "#d35e13", "#417246", "#2274A5"),
  lwd = 1,
  ...
)
```

## Arguments

| | |
|---|---|
| `x` | An object returned by `stl()`. |
| `plot_type` | Either `"multiple"` for one panel per component or `"single"` for one plot containing all components. |
| `multiplicative` | Logical. Is the decomposition multiplicative? If `NULL`, the method looks for `x$call$multiplicative`. If this is not available, as in older `stlpro` objects, an additive decomposition is assumed. |
| `xlab` | Label for the x-axis. |
| `ylab` | Label for the y-axis of the observed-data panel. |
| `main` | Optional overall plot title. |
| `labels` | Optional character vector of length four giving labels for the observed, seasonal, trend, and remainder panels. |
| `col` | Colours used for the observed, seasonally adjusted, seasonal, trend, and remainder series, respectively. |
| `lwd` | Line width. |
| `...` | Further graphical parameters passed to `graphics::plot()`. |

### Value

Invisibly returns `x`.

---

`print.cv_results` *Print method for cross-validation results*

---

### Description

Pretty-prints an object of class `"cv_results"`.

### Usage

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

### Arguments

| | |
|---|---|
| `x` | Object of class `"cv_results"` as returned by `run_cross_validation`. |
| `...` | Currently ignored. |

### Value

Invisibly returns the printed character string.

### Author(s)

Daniel Ollech

---

`stl` *Enhanced STL Decomposition*

---

### Description

Decompose a time series into seasonal, trend, and remainder components using STL, optionally with robust fitting. Compare to stats::stl it included cross-validation, multiplicative decompositions, allows non-integer seasonal periods and can handle missing values.

## Usage

```
stl(
  y,
  period = NULL,
  multiplicative = FALSE,
  s_window = NULL,
  t_window = NULL,
  l_window = NULL,
  s_degree = 0,
  t_degree = 1,
  l_degree = 1,
  s_jump = NULL,
  t_jump = NULL,
  l_jump = NULL,
  inner = NULL,
  outer = 0,
  robust = NULL,
  s_window_candidates = c(7, 9, 13, 19, 37),
  error_metric = "RMSE"
)
```

## Arguments

| | |
|---|---|
| y | Numeric vector or time series. |
| period | Seasonal period, can be non-integer. If NULL and y is ts, uses frequency(y). |
| multiplicative | Logical. If TRUE a multiplicative decomposition is calculated, additive otherwise. |
| s_window | Seasonal window (odd integer >= 3). If NULL, is set using cross-validation. |
| t_window | Trend window (odd integer >= 3). If NULL, uses the default, namely nextodd(ceiling((1.5*period) / (1-(1.5/s_window)))). |
| l_window | Low-pass window (odd integer >= 3). If NULL, uses default. |
| s_degree | Degree of locally-fitted polynomial in seasonal smoothing (0 or 1). |
| t_degree | Degree of locally-fitted polynomial in trend smoothing (0 or 1). |
| l_degree | Degree of locally-fitted polynomial in low-pass smoothing (0 or 1). |
| s_jump | Seasonal jump (integer >= 1). If NULL, uses default. |
| t_jump | Trend jump (integer >= 1). If NULL, uses default. |
| l_jump | Low-pass jump (integer >= 1). If NULL, uses default. |
| inner | Number of inner iterations. If NULL, uses a default determined by outer: one inner iteration when outer > 0, otherwise two. Ignored when robust is not NULL. |
| outer | Number of outer robustifying iterations (default 0). Ignored when robust is not NULL. |

| | |
|---|---|
| robust | Logical scalar controlling the robustness iteration settings. If TRUE, robust fitting is used with inner = 1 and outer = 15. If FALSE, non-robust fitting is used with inner = 2 and outer = 0. If NULL (the default), inner and outer are used as supplied. |
| s_window_candidates | Candidates for cross-validation based selection of s_window. |
| error_metric | Error metric used for cross-validation: "MAE", "MSE" or "RMSE". |

## Details

Values of period that are within 1e-8 of an integer are rounded to that integer. Values must be finite scalars greater than 1.

## Value

A list with components: seasonal, trend, remainder, sa, weights, call, cv_results.

## Author(s)

Daniel Ollech

# Index

graphics::plot(), *2*

plot.stlpro, 2
print.cv_results, 3

run_cross_validation, *3*

stl, 3
stl(), *2*