Package: mhpfilter (via r-universe)

May 15, 2026

Type Package

Title Modified Hodrick-Prescott Filter with Optimal Smoothing Parameter Selection

Version 0.1.0

Maintainer Muhammad Yaseen <myaseen208@gmail.com>

Description High-performance implementation of the Modified Hodrick-Prescott (HP) Filter for decomposing macroeconomic time series into trend and cyclical components. Based on the methodology of Choudhary, Hanif and Iqbal (2014) <doi:10.1080/00036846.2014.894631> ``On smoothing macroeconomic time series using the modified HP filter'', which uses generalized cross-validation (GCV) to automatically select the optimal smoothing parameter lambda, following McDermott (1997) ``An automatic method for choosing the smoothing parameter in the HP filter'' (as described in Coe and McDermott (1997) <doi:10.2307/3867497>). Unlike the standard HP filter that uses fixed lambda values (1600 for quarterly, 100 for annual data), this package estimates series-specific lambda values that minimize the GCV criterion. Implements efficient C++ routines via 'RcppArmadillo' for fast computation, supports batch processing of multiple series, and provides comprehensive visualization tools using 'ggplot2'. Particularly useful for cross-country macroeconomic comparisons, business cycle analysis, and when the appropriate smoothing parameter is uncertain.

License MIT + file LICENSE

Encoding UTF-8

URL https://myaseen208.com/mhpfilter/

BugReports https://github.com/myaseen208/mhpfilter/issues

Depends R (>= 4.0.0)

Imports Rcpp (>= 1.0.0), RcppArmadillo (>= 0.12.0.0.0), data.table (>= 1.14.0), collapse (>= 2.0.0), ggplot2 (>= 3.4.0)

LinkingTo Rcpp, RcppArmadillo

Suggests knitr (>= 1.40), rmarkdown (>= 2.20), testthat (>= 3.0.0), tidyverse (>= 2.0.0), fastverse (>= 0.3.0)

VignetteBuilder knitr

RoxygenNote 7.3.3

NeedsCompilation yes

Language en-US

Repository https://myaseen208.r-universe.dev

Date/Publication 2026-02-13 07:41:45 UTC

RemoteUrl https://github.com/myaseen208/mhpfilter

RemoteRef HEAD

RemoteSha 308f6886e1eadc3ec9a8e52f3bb799facac3c209

Contents

`mhpfilter-package` *mhpfilter: Fast Modified Hodrick-Prescott Filter*

Description

High-performance implementation of the Modified HP Filter for decomposing time series into trend and cyclical components. Based on the methodology of Choudhary, Hanif & Iqbal (2014) which uses generalized cross-validation to automatically select the optimal smoothing parameter lambda.

Details

The standard Hodrick-Prescott (1997) filter decomposes a time series y_t into trend g_t and cycle c_t components by minimizing:

$$\sum_{t=1}^{T}(y_t - g_t)^2 + \lambda \sum_{t=1}^{T-2}[(g_{t+2} - g_{t+1}) - (g_{t+1} - g_t)]^2$$

where λ is the smoothing parameter that controls the trade-off between trend smoothness and cycle fit.

The Modified HP Filter (McDermott, 1997) selects λ optimally using generalized cross-validation (GCV). The GCV criterion is:

$$GCV(\lambda) = \frac{SSR(\lambda)}{T}\left[1 + \frac{2}{T\lambda}\right]$$

where $SSR(\lambda)$ is the sum of squared residuals. The optimal λ minimizes this GCV criterion.

Main Functions

- `mhp_filter`: Apply Modified HP filter to single series
- `hp_filter`: Apply standard HP filter with fixed lambda
- `mhp_batch`: Batch process multiple series efficiently
- `mhp_compare`: Compare HP vs Modified HP for single series
- `batch_compare`: Compare HP vs Modified HP for multiple series
- `autoplot.mhp`: ggplot2 visualization for mhp objects
- `plot_comparison`: Compare HP and Modified HP visually
- `plot_batch`: Visualize batch processing results
- `get_lambda`: Extract optimal lambda from results
- `get_gcv`: Extract GCV value from results

Author(s)

Maintainer: Muhammad Yaseen <myaseen208@gmail.com> (ORCID)

Other contributors:

- Javed Iqbal <Javed.iqbal6@sbp.org.pk> (Original methodology author) [contributor]
- M. Nadim Hanif <Nadeem.hanif@sbp.org.pk> (Original methodology author) [contributor]

References

Choudhary, M.A., Hanif, M.N., & Iqbal, J. (2014). On smoothing macroeconomic time series using the modified HP filter. *Applied Economics*, 46(19), 2205-2214.

Hodrick, R.J., & Prescott, E.C. (1997). Postwar US business cycles: An empirical investigation. *Journal of Money, Credit and Banking*, 29(1), 1-16.

McDermott, C.J. (1997). Note on the modified Hodrick-Prescott filter. *IMF Working Paper No. 97/108*.

See Also

Useful links:

- https://myaseen208.com/mhpfilter/
- Report bugs at https://github.com/myaseen208/mhpfilter/issues

---

`autoplot.mhp` *Plot Method for mhp Objects*

Description

Create a publication-quality ggplot2 visualization of Modified HP filter results.

Usage

```
## S3 method for class 'mhp'
autoplot(object, ...)
```

Arguments

`object`	An object of class "mhp" from `mhp_filter(as_dt = FALSE)`.
`...`	Additional arguments passed to ggplot2 functions.

Details

Creates a three-panel plot showing: 1. Original series with trend overlay 2. Trend component 3. Cyclical component

The plot includes optimal lambda and GCV in the title, and uses consistent formatting suitable for publications.

Value

A ggplot object.

Examples

```
set.seed(42)
n <- 120
# Create a realistic macroeconomic series
trend <- cumsum(c(0, rnorm(n - 1, mean = 0.5, sd = 0.3)))
cycle <- 3 * sin(2 * pi * (1:n) / 30) + rnorm(n, sd = 0.8)
y <- trend + cycle + 100 # Add level for realism

result <- mhp_filter(y, max_lambda = 10000, as_dt = FALSE)

if (require(ggplot2)) {
  # Basic plot
  autoplot(result)
```

```
  # Customized plot
  p <- autoplot(result)
  p <- p +
    ggplot2::theme(
      plot.title = ggplot2::element_text(size = 14, face = "bold"),
      strip.text = ggplot2::element_text(size = 12, face = "bold")
    ) +
    ggplot2::labs(caption = "Data: Simulated macroeconomic series")
  print(p)
}
```

`batch_compare` *Batch Comparison of HP vs Modified HP*

Description

Compare HP and Modified HP filters across multiple time series. Useful for panel data analysis and method validation.

Usage

```
batch_compare(X, frequency = c("quarterly", "annual"), max_lambda = 100000L)
```

Arguments

`X`	Matrix or data.frame. Each column is a separate time series.
`frequency`	Character. Data frequency: "quarterly" or "annual".
`max_lambda`	Integer. Maximum lambda for Modified HP search. Default 100000.

Details

For each series in X, this function: 1. Applies standard HP filter with frequency-appropriate lambda 2. Applies Modified HP filter with GCV optimization 3. Calculates comparison statistics on cyclical components

The comparison helps identify: - Series where Modified HP substantially changes cycle properties - Optimal lambdas across different types of series - Relative performance of automatic vs fixed smoothing

Value

A data.table with comparison metrics for each series:

series Series identifier

hp_lambda Lambda used for HP filter (1600 or 100)

mhp_lambda Optimal lambda from Modified HP

hp_cycle_sd Cycle standard deviation (HP)

mhp_cycle_sd Cycle standard deviation (Modified HP)

sd_diff Difference in cycle SD (MHP - HP)

hp_ar1 Cycle AR(1) coefficient (HP)

mhp_ar1 Cycle AR(1) coefficient (Modified HP)

ar1_diff Difference in AR(1) (MHP - HP)

relative_sd mhp_cycle_sd / hp_cycle_sd

References

Choudhary, M.A., Hanif, M.N., & Iqbal, J. (2014). On smoothing macroeconomic time series using the modified HP filter. *Applied Economics*, 46(19), 2205-2214.

Examples

```
# Example 1: Country GDP comparison
set.seed(101)
n <- 80
countries <- c("USA", "UK", "Japan", "Germany", "France", "Italy", "Canada", "Australia")
gdp_data <- sapply(countries, function(ctry) {
  # Varying volatility and persistence
  vol <- runif(1, 0.5, 2.5)
  persist <- runif(1, 0.6, 0.95)
  trend <- cumsum(rnorm(n, 0.5, 0.3))
  cycle <- arima.sim(list(ar = persist), n, sd = vol)
  trend + cycle
})

results <- batch_compare(gdp_data, frequency = "quarterly", max_lambda = 10000)
print(results)

# Example 2: Sectoral analysis with visualization
set.seed(2024)
n_time <- 100
sectors <- c("Tech", "Finance", "Energy", "Healthcare", "Consumer")
sector_returns <- matrix(rnorm(n_time * length(sectors)), nrow = n_time)

# Add sector-specific characteristics
for (i in 1:length(sectors)) {
  drift <- runif(1, -0.1, 0.3)
  volatility <- runif(1, 0.5, 2.0)
  sector_returns[, i] <- cumsum(rnorm(n_time, mean = drift / 100, sd = volatility / 100)) +
    runif(1, 0.5, 2) * sin(2 * pi * (1:n_time) / (20 + i * 3))
}
colnames(sector_returns) <- sectors

sector_comparison <- batch_compare(sector_returns, frequency = "quarterly", max_lambda = 5000)

if (require(ggplot2)) {
  # Plot lambda comparison
```

```
  lambda_plot <- ggplot2::ggplot(
    sector_comparison,
    ggplot2::aes(x = series, y = mhp_lambda)
  ) +
    ggplot2::geom_col(fill = "steelblue", alpha = 0.7) +
    ggplot2::geom_hline(yintercept = 1600, linetype = "dashed", color = "red") +
    ggplot2::labs(
      title = "Modified HP Optimal Lambdas by Sector",
      subtitle = "Red line shows fixed HP lambda (1600)",
      x = "Sector", y = "Optimal Lambda"
    ) +
    ggplot2::theme_minimal() +
    ggplot2::theme(axis.text.x = ggplot2::element_text(angle = 45, hjust = 1))

  print(lambda_plot)
}
```

---

get_gcv *Extract GCV Value*

Description

Extract the generalized cross-validation (GCV) value from Modified HP filter results.

Usage

```
get_gcv(x)
```

Arguments

x	Result from mhp_filter (either data.table or mhp object).

Value

Numeric. The GCV value at the optimal lambda.

Examples

```
set.seed(123)
result <- mhp_filter(cumsum(rnorm(100)), max_lambda = 10000)
get_gcv(result)

# With mhp object
result_obj <- mhp_filter(cumsum(rnorm(100)), max_lambda = 10000, as_dt = FALSE)
get_gcv(result_obj)
```

get_lambda *Extract Optimal Lambda*

Description

Extract the optimal smoothing parameter lambda from Modified HP filter results.

Usage

```
get_lambda(x)
```

Arguments

x	Result from mhp_filter (either data.table or mhp object).

Value

Integer. The optimal lambda value.

Examples

```
set.seed(123)
result <- mhp_filter(cumsum(rnorm(100)), max_lambda = 10000)
get_lambda(result)

# With mhp object
result_obj <- mhp_filter(cumsum(rnorm(100)), max_lambda = 10000, as_dt = FALSE)
get_lambda(result_obj)
```

hp_filter *Standard Hodrick-Prescott Filter*

Description

Decomposes a time series into trend and cyclical components using the standard HP filter with a fixed smoothing parameter lambda.

Usage

```
hp_filter(x, lambda = 1600, as_dt = TRUE)
```

Arguments

x	Numeric vector. The time series to decompose.
lambda	Numeric. The smoothing parameter. Default 1600 (quarterly data). Common values: 1600 (quarterly), 100 (annual), 14400 (monthly).
as_dt	Logical. If TRUE (default), returns a data.table. If FALSE, returns a list.

Details

The HP filter solves the minimization problem:

$$\min_{\{g_t\}} \left\{ \sum_{t=1}^{T} (y_t - g_t)^2 + \lambda \sum_{t=2}^{T-1} [(g_{t+1} - g_t) - (g_t - g_{t-1})]^2 \right\}$$

The solution is obtained by solving:

$$(I + \lambda K'K)g = y$$

where K is the second-difference matrix.

Value

If `as_dt = TRUE`: A `data.table` with columns:

original The input series

trend The estimated trend component

cycle The cyclical component

With attribute `lambda` (the input lambda value).

If `as_dt = FALSE`: A list containing `original`, `trend`, `cycle`, and `lambda`.

References

Hodrick, R.J., & Prescott, E.C. (1997). Postwar US business cycles: An empirical investigation. *Journal of Money, Credit and Banking*, 29(1), 1-16.

Ravn, M.O., & Uhlig, H. (2002). On adjusting the Hodrick-Prescott filter for the frequency of observations. *Review of Economics and Statistics*, 84(2), 371-376.

Examples

```
# Example 1: Simple random walk with cycle
set.seed(123)
n <- 80
y <- cumsum(rnorm(n)) + sin((1:n) * pi / 10)
result <- hp_filter(y, lambda = 1600)
head(result)

# Example 2: GDP-like series
set.seed(456)
gdp <- cumsum(rnorm(100, mean = 0.5, sd = 0.3)) + 2 * cos(2 * pi * (1:100) / 40)
gdp_decomp <- hp_filter(gdp, lambda = 1600)

# Plot the decomposition
if (require(ggplot2)) {
  plot_data <- data.table::data.table(
    t = 1:length(gdp),
    Original = gdp,
```

```
  Trend = gdp_decomp$trend,
  Cycle = gdp_decomp$cycle
)
plot_data_long <- data.table::melt(plot_data, id.vars = "t")

ggplot2::ggplot(plot_data_long, ggplot2::aes(x = t, y = value, color = variable)) +
  ggplot2::geom_line(linewidth = 0.8) +
  ggplot2::facet_wrap(~variable, ncol = 1, scales = "free_y") +
  ggplot2::labs(
    title = "HP Filter Decomposition (lambda = 1600)",
    x = "Time", y = "Value"
  ) +
  ggplot2::theme_minimal() +
  ggplot2::theme(legend.position = "none")
}
```

`mhp_batch` *Batch Modified HP Filter*

Description

Process multiple time series efficiently using the Modified HP Filter. Optimized for processing large collections of time series (e.g., panel data).

Usage

```
mhp_batch(X, max_lambda = 100000L)
```

Arguments

`X`	Matrix or data.frame. Each column is a separate time series. Rows represent time periods, columns represent different series.
`max_lambda`	Integer. Maximum lambda value to search. Default 100000.

Details

This function efficiently processes multiple series by: 1. Pre-computing the $K'K$ matrix once for all series 2. Performing parallelizable grid search for each series 3. Using optimized C++ routines via RcppArmadillo

Value

A data.table in long format with columns:

series Series identifier (column name or V1, V2, ...)

t Time index (1, 2, ..., T)

original Original series values

trend Estimated trend component

cycle Cyclical component

With attribute `"lambdas"` containing a data.table of optimal lambdas and GCV values for each series:

series Series identifier

lambda Optimal lambda for the series

gcv GCV value at optimal lambda

Examples

```
# Example 1: Multiple macroeconomic series
set.seed(456)
n_periods <- 60
n_countries <- 5
gdp_matrix <- matrix(nrow = n_periods, ncol = n_countries)
colnames(gdp_matrix) <- c("USA", "UK", "Germany", "France", "Japan")

# Generate series with different characteristics
for (i in 1:n_countries) {
  trend <- cumsum(rnorm(n_periods, mean = 0.5, sd = 0.3))
  cycle <- rnorm(n_periods, sd = 1 + i * 0.2) # Increasing volatility
  gdp_matrix[, i] <- trend + cycle
}

# Apply batch Modified HP filter
results <- mhp_batch(gdp_matrix, max_lambda = 10000)

# Extract optimal lambdas
lambdas <- attr(results, "lambdas")
print(lambdas)

# Example 2: Sectoral data
set.seed(789)
n_time <- 120
n_sectors <- 8
sector_names <- c(
  "Agriculture", "Mining", "Manufacturing", "Construction",
  "Trade", "Transport", "Finance", "Services"
)
sector_data <- matrix(rnorm(n_time * n_sectors), nrow = n_time)

# Add sector-specific trends and cycles
for (i in 1:n_sectors) {
  trend_growth <- runif(1, 0.2, 1.0)
  cycle_amplitude <- runif(1, 0.5, 3.0)
  sector_data[, i] <- cumsum(rnorm(n_time, mean = trend_growth / 4, sd = 0.3)) +
    cycle_amplitude * sin(2 * pi * (1:n_time) / (20 + i * 5))
}
colnames(sector_data) <- sector_names
```

```
sector_results <- mhp_batch(sector_data, max_lambda = 50000)

# View results for first few periods
head(sector_results)
```

`mhp_compare` *Compare HP vs Modified HP Filter*

Description

Compare the standard HP filter with the Modified HP filter for a single series. Provides summary statistics for both methods including cycle properties.

Usage

```
mhp_compare(x, frequency = c("quarterly", "annual"), max_lambda = 100000L)
```

Arguments

`x`	Numeric vector. The time series to analyze.
`frequency`	Character. Data frequency: "quarterly" (lambda=1600) or "annual" (lambda=100).
`max_lambda`	Integer. Maximum lambda for Modified HP search. Default 100000.

Details

The comparison includes: 1. Standard HP filter with fixed lambda (1600 for quarterly, 100 for annual) 2. Modified HP filter with GCV-optimized lambda

Statistics calculated on the cyclical component help assess filter performance: - Lower cycle SD suggests smoother trend - AR(1) near 0 suggests successful cycle extraction - Near-zero mean suggests proper centering

Value

A data.table with comparison statistics for both methods:

method "HP" or "Modified HP"

lambda Smoothing parameter used

cycle_sd Standard deviation of cyclical component

cycle_mean Mean of cyclical component

ar1 First-order autocorrelation of cyclical component

cycle_range Range of cyclical component (max - min)

gcv GCV value (NA for standard HP)

Examples

```
# Example 1: Quarterly GDP-like series
set.seed(789)
n <- 100
gdp <- cumsum(rnorm(n, mean = 0.7, sd = 0.5)) + 2 * cos(2 * pi * (1:n) / 32)
comparison <- mhp_compare(gdp, frequency = "quarterly", max_lambda = 10000)
print(comparison)

# Example 2: Annual series
set.seed(101)
n_annual <- 50
annual_series <- cumsum(rnorm(n_annual, mean = 2.0, sd = 1.0)) +
  3 * sin(2 * pi * (1:n_annual) / 10)
annual_comparison <- mhp_compare(annual_series, frequency = "annual", max_lambda = 5000)
print(annual_comparison)

# Example 3: Visual comparison
set.seed(2023)
test_series <- cumsum(rnorm(120, mean = 0.5, sd = 0.4)) +
  runif(1, 1, 3) * sin(2 * pi * (1:120) / 30)

comp_result <- mhp_compare(test_series, frequency = "quarterly", max_lambda = 20000)

if (require(ggplot2)) {
  # Create visualization
  hp_result <- hp_filter(test_series, lambda = 1600, as_dt = FALSE)
  mhp_result <- mhp_filter(test_series, max_lambda = 20000, as_dt = FALSE)

  plot_comparison(test_series, frequency = "quarterly", max_lambda = 20000)
}
```

`mhp_filter` *Modified Hodrick-Prescott Filter*

Description

Decomposes a time series into trend and cyclical components using the Modified HP Filter, which automatically selects the optimal smoothing parameter lambda via generalized cross-validation (GCV).

Usage

```
mhp_filter(x, max_lambda = 100000L, as_dt = TRUE)
```

Arguments

`x`	Numeric vector. The time series to decompose. Must have at least 5 observations and no missing values.

`max_lambda`	Integer. Maximum lambda value to search over. Default is 100000, which covers most macroeconomic applications. The search ranges from 1 to 'max_lambda'.
`as_dt`	Logical. If TRUE (default), returns a data.table. If FALSE, returns a list with class "mhp".

Details

The function performs a grid search over lambda values from 1 to 'max_lambda' and selects the lambda that minimizes the GCV criterion. For each lambda, it solves the system:

$$(I + \lambda K'K)g = y$$

where K is the second-difference matrix, g is the trend, and y is the original series.

Value

If `as_dt = TRUE`: A `data.table` with columns:

original The input series

trend The estimated trend component

cycle The cyclical component (original - trend)

With attributes `lambda` (optimal lambda) and `gcv` (GCV value).

If `as_dt = FALSE`: A list with class "mhp" containing elements:

original The input series

trend The estimated trend component

cycle The cyclical component

lambda Optimal smoothing parameter

gcv Generalized cross-validation value

References

Choudhary, M.A., Hanif, M.N., & Iqbal, J. (2014). On smoothing macroeconomic time series using the modified HP filter. *Applied Economics*, 46(19), 2205-2214.

See Also

`hp_filter`, `autoplot.mhp`, `get_lambda`, `get_gcv`

Examples

```
# Simulate a trend + cycle series
set.seed(42)
n <- 100
trend <- cumsum(c(0, rnorm(n - 1, mean = 0.5, sd = 0.2)))
cycle <- 2 * sin(2 * pi * (1:n) / 20) + rnorm(n, sd = 0.5)
y <- trend + cycle
```

```
# Apply Modified HP filter
result <- mhp_filter(y, max_lambda = 10000)

# Extract optimal lambda
get_lambda(result)

# Extract GCV value
get_gcv(result)

# Print summary
print(result)

# Plot with ggplot2
if (require(ggplot2)) {
  autoplot(mhp_filter(y, max_lambda = 10000, as_dt = FALSE))
}
```

`plot_batch` *Plot Batch Results*

Description

Create a ggplot2 visualization of batch filter results. Useful for comparing multiple series' cyclical components or trends.

Usage

```
plot_batch(x, show = c("cycle", "trend"), facet = TRUE, highlight = NULL)
```

Arguments

`x`	Result from `mhp_batch`.
`show`	Character. What to show: "cycle" (default) or "trend".
`facet`	Logical. If TRUE, use faceting; if FALSE, overlay series.
`highlight`	Character vector. Names of series to highlight (others shown faintly).

Details

Creates visualizations for batch processing results: - Cycle plot: Shows business cycle components across series - Trend plot: Shows trend components across series

Options for faceting or overlay, with highlighting capability.

Value

A ggplot object.

Examples

```
set.seed(456)
# Create multi-country dataset
n_time <- 80
countries <- c("USA", "UK", "Germany", "France", "Japan", "Canada")
gdp_data <- matrix(nrow = n_time, ncol = length(countries))

for (i in seq_along(countries)) {
  # Different growth rates and cycle patterns
  growth <- runif(1, 0.3, 1.0)
  cycle_freq <- 20 + runif(1, -5, 15)
  cycle_amp <- runif(1, 0.5, 2.5)

  gdp_data[, i] <- 100 + cumsum(rnorm(n_time, mean = growth / 100, sd = 0.4 / 100)) +
    cycle_amp * sin(2 * pi * (1:n_time) / cycle_freq)
}
colnames(gdp_data) <- countries

results <- mhp_batch(gdp_data, max_lambda = 10000)

if (require(ggplot2)) {
  # Show cycles with faceting
  plot_batch(results, show = "cycle", facet = TRUE)

  # Show trends overlaid
  plot_batch(results, show = "trend", facet = FALSE)

  # Highlight specific countries
  plot_batch(results, show = "cycle", facet = FALSE, highlight = c("USA", "Germany"))

  # Customized plot
  p <- plot_batch(results, show = "cycle", facet = TRUE)
  p <- p +
    ggplot2::labs(
      title = "Business Cycle Components: Selected Countries",
      subtitle = "Modified HP Filter Decomposition"
    ) +
    ggplot2::theme(
      strip.text = ggplot2::element_text(face = "bold", size = 9),
      axis.text.x = ggplot2::element_text(angle = 45, hjust = 1)
    )
  print(p)
}
```

`plot_comparison` *Plot Comparison of HP vs Modified HP*

Description

Create a ggplot2 comparison of HP and Modified HP filter trends. Useful for visualizing differences between fixed and optimized smoothing.

Usage

```
plot_comparison(
  x,
  frequency = c("quarterly", "annual"),
  max_lambda = 100000L,
  show_cycle = FALSE
)
```

Arguments

x	Numeric vector. The time series.
frequency	Character. Data frequency: "quarterly" or "annual".
max_lambda	Integer. Maximum lambda for Modified HP search.
show_cycle	Logical. If TRUE, also show cyclical components.

Details

Creates comparison plots showing: 1. Original series with HP and Modified HP trends overlaid 2. (Optional) Cyclical components from both methods

The plot uses distinct colors and line styles to differentiate methods, with annotations showing lambda values.

Value

A ggplot object.

Examples

```
set.seed(123)
# Simulate realistic economic data
n <- 100
base_level <- 100
growth_rate <- 0.5
volatility <- 1.2

y <- base_level + cumsum(rnorm(n, mean = growth_rate / 100, sd = volatility / 100)) +
  2.5 * sin(2 * pi * (1:n) / 25) + rnorm(n, sd = 0.5)

if (require(ggplot2)) {
  # Basic comparison
  plot_comparison(y, frequency = "quarterly", max_lambda = 10000)

  # With cycles
  plot_comparison(y, frequency = "quarterly", max_lambda = 10000, show_cycle = TRUE)
```

```
  # Customized plot
  p <- plot_comparison(y, frequency = "quarterly", max_lambda = 10000)
  p <- p +
    ggplot2::labs(
      title = "HP vs Modified HP: Trend Comparison",
      subtitle = "Quarterly macroeconomic series"
    ) +
    ggplot2::theme(
      plot.title = ggplot2::element_text(face = "bold", size = 14),
      legend.title = ggplot2::element_blank(),
      legend.position = "bottom"
    )
  print(p)
}
```

`print.mhp` *Print Method for mhp Objects*

Description

Print a summary of Modified HP filter results.

Usage

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

Arguments

`x`	An object of class "mhp" from `mhp_filter(as_dt = FALSE)`.
`...`	Additional arguments (ignored).

Details

Prints a formatted summary including: - Number of observations - Optimal lambda value - GCV criterion value - Cycle statistics (mean, SD, AR1)

Suitable for quick inspection of filter results.

Value

Invisibly returns the input object.

Examples

```
set.seed(42)
y <- cumsum(rnorm(100)) + sin((1:100) * pi / 20)
result <- mhp_filter(y, max_lambda = 10000, as_dt = FALSE)
print(result)
```

Index