

Seasonality Detection Methods: A Comparative Study

fdars Package

today

Contents

Introduction	1
Detection Methods	2
AIC Comparison (Fourier vs P-spline)	2
FFT Confidence	2
ACF Confidence	2
Variance Strength	3
Spectral Strength	3
Simulation Studies	3
Simulation 1: Varying Seasonal Strength	3
Setup	3
Code	4
Configuration	4
Detection thresholds (calibrated to ~5% FPR on pure noise)	4
Generate seasonal curve	4
Results	6
F1 Score vs Trend Strength	6
False Positive Rate by Trend Strength	7
Simulation 3: Multiple Trend Types	7
Setup	7
Code	7
Threshold Guidelines	9
Conclusion	9
Appendix: File Listing	9

Introduction

This report describes and compares five methods for detecting seasonality in functional time series data. We evaluate each method's performance across different scenarios including varying seasonal strengths, non-linear trends, and different trend types.

The goal is to answer: **Given a time series, how can we reliably determine if it contains a seasonal pattern?**

Detection Methods

AIC Comparison (Fourier vs P-spline)

Concept: If data is seasonal, a Fourier basis should fit better than P-splines because Fourier bases naturally capture periodic patterns.

Mathematical formulation:

For a curve $y(t)$, we fit two models:

1. **Fourier basis:** $\hat{y}(t) = \sum_{k=0}^K a_k \cos(2\pi kt) + b_k \sin(2\pi kt)$
2. **P-spline:** $\hat{y}(t) = \sum_{j=1}^J c_j B_j(t)$ with penalty $\lambda \int [\hat{y}''(t)]^2 dt$

We compute AIC for each:

$$\text{AIC} = n \log(\text{RSS}/n) + 2 \cdot \text{edf}$$

where RSS is the residual sum of squares and edf is the effective degrees of freedom.

Detection rule: Seasonality detected if $\text{AIC}_{\text{P-spline}} - \text{AIC}_{\text{Fourier}} > 0$

Interpretation: When Fourier has lower AIC, the periodic structure is significant enough to justify the global periodic assumption over the local flexibility of splines.

FFT Confidence

Concept: Use Fast Fourier Transform to detect dominant frequencies. Strong peaks in the periodogram indicate periodic components.

Mathematical formulation:

Given a time series y_1, y_2, \dots, y_n , compute the discrete Fourier transform:

$$Y_k = \sum_{j=1}^n y_j e^{-2\pi i(j-1)(k-1)/n}$$

The periodogram (power spectrum) is:

$$P_k = |Y_k|^2$$

Detection score:

$$\text{Confidence} = \frac{\max_k P_k}{\text{mean}(P_k)}$$

Detection rule: Seasonality detected if Confidence > 6.0

Interpretation: A high ratio indicates one frequency dominates, suggesting periodicity rather than random noise.

ACF Confidence

Concept: Autocorrelation at the seasonal lag should be high for seasonal data.

Mathematical formulation:

The autocorrelation function at lag h is:

$$\rho_h = \frac{\sum_{t=1}^{n-h} (y_t - \bar{y})(y_{t+h} - \bar{y})}{\sum_{t=1}^n (y_t - \bar{y})^2}$$

For seasonal data with period p , we expect ρ_p to be significantly positive.

Detection rule: Seasonality detected if ACF confidence > 0.25

Interpretation: High autocorrelation at the seasonal lag indicates the pattern repeats.

Variance Strength

Concept: Decompose variance into seasonal and residual components. High seasonal variance ratio indicates seasonality.

Mathematical formulation:

Decompose the series: $y_t = T_t + S_t + R_t$ (trend + seasonal + residual)

The seasonal strength is:

$$SS_{\text{var}} = 1 - \frac{\text{Var}(R_t)}{\text{Var}(y_t - T_t)}$$

Detection rule: Seasonality detected if $SS_{\text{var}} > 0.2$

Interpretation: Values close to 1 mean the seasonal component dominates; values close to 0 mean residual noise dominates.

Important: The `period` parameter must be in the same units as `argvals`. For data normalized to $[0,1]$ with 5 annual cycles, use `period = 0.2`.

Spectral Strength

Concept: Measure the proportion of spectral power at the seasonal frequency.

Mathematical formulation:

Using the periodogram P_k , identify the seasonal frequency $f_s = 1/\text{period}$.

$$SS_{\text{spectral}} = \frac{\sum_{k \in \mathcal{S}} P_k}{\sum_k P_k}$$

where \mathcal{S} includes the seasonal frequency and its harmonics.

Detection rule: Seasonality detected if $SS_{\text{spectral}} > 0.3$

Interpretation: High values indicate spectral energy is concentrated at seasonal frequencies.

Simulation Studies

Simulation 1: Varying Seasonal Strength

Setup

This simulation tests how well each method detects seasonality at different signal strengths.

Parameters:

- 11 seasonal strength levels: 0.0, 0.1, ..., 1.0
- 50 curves per strength level
- 5 years of monthly data (60 observations)
- Noise standard deviation: 0.3

Signal model:

$$y(t) = s \cdot [\sin(2\pi \cdot 5t) + 0.3 \cos(4\pi \cdot 5t)] + \epsilon, \quad \epsilon \sim N(0, 0.3^2)$$

where s is the seasonal strength ($0 = \text{no seasonality}$, $1 = \text{full seasonality}$).

Ground truth: A curve is classified as “truly seasonal” if $s \geq 0.2$.

Code

```
“{r} #| eval: false #| code-fold: true #| code-summary: “Show simulation code”  
library(fdars) library(ggplot2) library(tidyr) library(dplyr)  
set.seed(42)
```

Configuration

```
n_strengths <- 11 n_curves_per_strength <- 50 n_years <- 5 n_months <- n_years * 12 noise_sd <- 0.3
```

Detection thresholds (calibrated to ~5% FPR on pure noise)

```
detection_thresholds <- list( aic_comparison = 0, fft_confidence = 6.0, acf_confidence = 0.25,  
strength_variance = 0.2, strength_spectral = 0.3 )  
seasonal_strengths <- seq(0, 1, length.out = n_strengths) t <- seq(0, 1, length.out = n_months)
```

Generate seasonal curve

```
generate_seasonal_curve <- function(t, strength, noise_sd = 0.3) { n_cycles <- length(t) / 12 seasonal <-  
strength * sin(2 * pi * n_cycles * t) seasonal <- seasonal + strength * 0.3 * cos(4 * pi * n_cycles * t) noise  
<- rnorm(length(t), sd = noise_sd) return(seasonal + noise) }
```

Results

```
![Detection rates by seasonal strength](seasonality_detection_comparison.pdf){#fig-detection-rates width=
```

****How to interpret**:**

- The x-axis shows the true seasonal strength ($0 = \text{pure noise}$, $1 = \text{strong seasonality}$)
- The y-axis shows what percentage of curves each method classified as "seasonal"
- The vertical dashed line at 0.2 marks the ground truth threshold
- ****Ideal behavior**:** 0% detection below the threshold, 100% above

Classification Performance

Method	F1 Score	Precision	Recall	FPR	Specificity
Variance Strength	**97.3%**	98.2%	96.4%	2.0%	92.0%
Spectral Strength	95.3%	97.4%	93.3%	10.0%	89.0%
FFT Confidence	94.8%	99.3%	90.7%	4.0%	97.0%
AIC Comparison	91.5%	94.3%	88.9%	18.0%	76.0%
ACF Confidence	85.4%	98.3%	75.6%	10.0%	94.0%

****How to interpret**:**

- **F1 Score**: Harmonic mean of precision and recall (higher is better)
- **Precision**: Of curves detected as seasonal, what % are truly seasonal?
- **Recall**: Of truly seasonal curves, what % did we detect?
- **FPR**: False Positive Rate - what % of non-seasonal curves were incorrectly flagged?

Precision-Recall Analysis

![Precision-Recall curves](seasonality_detection_details.pdf){#fig-pr-curves width=100%}

How to interpret:

- Curves closer to the top-right corner are better
- The diamond markers show the operating point at the default threshold
- AUC-PR (Area Under the PR Curve) summarizes overall performance

Simulation 2: Non-linear Trend {#sec-sim2}

Setup

This simulation tests robustness when non-linear trends are added to the seasonal signal.

Parameters:

- 6 seasonal strength levels × 6 trend strength levels
- 30 curves per combination
- Non-linear trend: quadratic + cubic + sigmoid components

Signal model:

$y(t) = \text{Trend}(t, \tau) + \text{Seasonal}(t, s) + \epsilon$

where τ is the trend strength and s is the seasonal strength.

Trend function:

$\text{Trend}(t, \tau) = \tau \cdot [2(t-0.5)^2 + 0.5(t-0.3)^3 + 0.3 \cdot \sigma(10(t-0.6)) - 0.5]$

Code

```
```{r}
#| eval: false
#| code-fold: true
#| code-summary: "Show trend simulation code"

Non-linear trend function
generate_nonlinear_trend <- function(t, trend_strength) {
 quadratic <- 2 * (t - 0.5)^2
 cubic <- 0.5 * (t - 0.3)^3
 sigmoid <- 1 / (1 + exp(-10 * (t - 0.6)))
 trend <- trend_strength * (quadratic + cubic + 0.3 * sigmoid - 0.5)
 return(trend)
}

Generate curve with trend + seasonal + noise
generate_curve <- function(t, seasonal_strength, trend_strength, noise_sd = 0.3) {
```

```

trend <- generate_nonlinear_trend(t, trend_strength)
n_cycles <- length(t) / 12
seasonal <- seasonal_strength * sin(2 * pi * n_cycles * t)
seasonal <- seasonal + seasonal_strength * 0.3 * cos(4 * pi * n_cycles * t)
noise <- rnorm(length(t), sd = noise_sd)
return(trend + seasonal + noise)
}

```

## Results



Figure 1: Detection rates heatmap by seasonal and trend strength

### How to interpret:

- Each cell shows the detection rate for a combination of seasonal strength (x) and trend strength (y)
- Blue = low detection rate, Red = high detection rate
- The dashed line separates non-seasonal (left) from seasonal (right) ground truth

### F1 Score vs Trend Strength

Method	No Trend	Max Trend	F1 Drop
Spectral	96.3%	92.5%	3.9%
FFT	93.7%	91.8%	2.0%
AIC	92.2%	87.0%	5.7%
ACF	87.4%	83.5%	4.5%

How to interpret:

- **F1 Drop:** How much performance degrades when strong trends are present
- Lower drop = more robust to trends

## False Positive Rate by Trend Strength

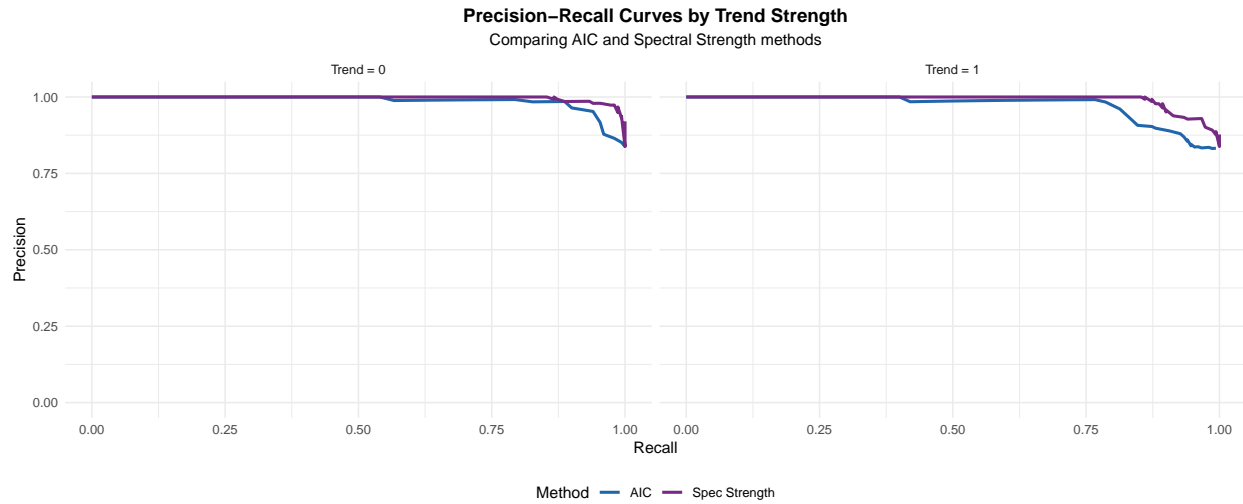


Figure 2: FPR when no seasonality is present, across trend strengths

**Key finding:** FFT’s FPR increases dramatically with trend strength because non-linear trends can create spurious peaks in the periodogram.

## Simulation 3: Multiple Trend Types

### Setup

This simulation tests which types of trends cause the most problems for each detection method.

### Trend types tested:

1. **None:** Flat baseline
2. **Linear:**  $f(t) = t - 0.5$
3. **Quadratic:**  $f(t) = (t - 0.5)^2 - 0.25$
4. **Cubic:**  $f(t) = 2(t - 0.5)^3$
5. **Exponential:**  $f(t) = e^{2t}/e^2 - 0.5$
6. **Logarithmic:**  $f(t) = \log(t + 0.1)$  (normalized)
7. **Sigmoid:**  $f(t) = 1/(1 + e^{-10(t-0.5)}) - 0.5$
8. **Slow sine:**  $f(t) = \sin(2\pi t)$  — one cycle over the entire series

### Code

“{r} #| eval: false #| code-fold: true #| code-summary: “Show trend types code”

```
trend_functions <- list(none = function(t, strength) rep(0, length(t)), linear = function(t, strength) strength
* (t - 0.5), quadratic = function(t, strength) strength * ((t - 0.5)^2 - 0.25), cubic = function(t, strength)
strength * 2 * (t - 0.5)^3, exponential = function(t, strength) strength * (exp(2 * t) / exp(2) - 0.5), logarithmic
= function(t, strength) { strength * (log(t + 0.1) - log(0.1)) / (log(1.1) - log(0.1)) - 0.5 * strength }, sigmoid
= function(t, strength) strength * (1 / (1 + exp(-10 * (t - 0.5)))) - 0.5, slow_sine = function(t, strength)
strength * sin(2 * pi * t))
```

### ### Results

![F1 scores by trend type](seasonality\_detection\_trend\_types\_f1.pdf){#fig-trend-types-f1 width=100%}

### ### FPR by Trend Type

![Example trend functions](seasonality\_detection\_trend\_types\_examples.pdf){#fig-trend-examples width=100%}

![FPR across trend types and strengths](seasonality\_detection\_trend\_types\_fpr.pdf){#fig-trend-types-fpr width=100%}

**\*\*How to interpret\*\*:**

- Each panel shows one trend type
- Lines show how FPR changes as trend strength increases
- **\*\*Slow sine is catastrophic for FFT\*\***: 100% FPR because FFT detects the slow oscillation as "seasonal"

### ### Most Problematic Trend Types

Trend Type	FFT FPR	Spectral FPR	Issue
slow_sine	100%	0%	FFT detects non-seasonal oscillation
quadratic	10%	5%	Minor
sigmoid	5%	5%	Minor
linear	0%	10%	Handled well

### # Key Findings

#### ## Method Ranking

1. **\*\*Variance Strength\*\*** (F1=97.3%, FPR=2%): Best overall when period is known
2. **\*\*Spectral Strength\*\*** (F1=95.3%, FPR=10%): Most robust to different trend types
3. **\*\*FFT Confidence\*\*** (F1=94.8%, FPR=4%): Good but vulnerable to slow oscillations
4. **\*\*AIC Comparison\*\*** (F1=91.5%, FPR=18%): Interpretable but higher FPR
5. **\*\*ACF Confidence\*\*** (F1=85.4%, FPR=10%): Conservative, misses weak seasonality

#### ## Critical Issues Found

1. **\*\*Period units matter\*\***: The `period` parameter in `seasonal\_strength()` must be in argvals units, not years
2. **\*\*FFT is vulnerable to slow oscillations\*\***: Any periodic signal (even non-seasonal) triggers detection

### # Recommendations

#### ## For Unknown Datasets

**\*\*Primary recommendation: Variance Strength\*\***

```
```{r}
```

```
#| eval: false
```

```
# Calculate period in argvals units
```

```
# If argvals is in [0,1] and you expect 5 annual cycles:
```

```
period_in_argvals_units <- 1 / 5 # = 0.2
```



```

strength <- seasonal_strength(fd,
                             period = period_in_argvals_units,
                             method = "variance",
                             detrend = "linear")
is_seasonal <- strength > 0.2

For robustness to unknown trends: Spectral Strength

“{r} #| eval: false

strength <- seasonal_strength(fd, period = period_in_argvals_units, method = “spectral”, detrend =
“linear”) is_seasonal <- strength > 0.3

**Ensemble approach (most robust)**:

```{r}
#| eval: false

var_detected <- seasonal_strength(fd, period, method = "variance") > 0.2
spec_detected <- seasonal_strength(fd, period, method = "spectral") > 0.3
fft_detected <- estimate_period(fd, method = "fft")$confidence > 6.0

Majority vote
is_seasonal <- (var_detected + spec_detected + fft_detected) >= 2

```

## Threshold Guidelines

Method	Threshold	Calibration
Variance Strength	0.2	95th percentile of noise ~0.17
Spectral Strength	0.3	95th percentile of noise ~0.29
FFT Confidence	6.0	95th percentile of noise ~5.7
ACF Confidence	0.25	95th percentile of noise ~0.22
AIC Difference	0	Fourier better → positive difference

## Conclusion

For detecting seasonality in functional time series:

1. **Variance Strength** is the most accurate method when the seasonal period is known
2. **Spectral Strength** is most robust to confounding trends and unknown oscillations
3. **FFT Confidence** works well but is vulnerable to slow non-seasonal oscillations
4. **AIC Comparison** provides an interpretable alternative but has higher false positive rates
5. **ACF Confidence** is conservative (low FPR) but misses weak seasonality

The key insight is that simple variance-based decomposition outperforms more complex spectral methods when properly configured with the correct period parameter.

## Appendix: File Listing

All simulation scripts and results are in `scripts/seasonal_simulation/`:

- `seasonality_detection_comparison.R` — Main comparison (Simulation 1)
- `seasonality_detection_with_trend.R` — Non-linear trend study (Simulation 2)

- `seasonality_detection_trend_types.R` — Multiple trend types (Simulation 3)
- `seasonal_basis_comparison.R` — AIC comparison study

PDF outputs contain the visualizations referenced in this report.