

Seasonality Detection Methods: A Comparative Study

fdars Package

2026-01-01

Table of contents

1	Executive Summary	2
1.1	Key Findings	2
1.2	Robustness to Real-World Challenges	3
1.3	Recommendations	3
1.3.1	Primary Recommendation: Use SAZED for Unknown Signals	3
1.3.2	When Period is Known: Use Variance Strength	3
1.3.3	When Trend is Present: Use CFDAutoperiod	4
1.3.4	Unified Detection Interface	4
1.3.5	Critical Notes	4
2	Introduction	5
3	Detection Methods	5
3.1	AIC Comparison (Fourier vs P-spline)	5
3.2	FFT Confidence	5
3.3	ACF Confidence	6
3.4	Variance Strength	6
3.5	Spectral Strength	7
3.6	Wavelet Strength	7
3.7	SAZED (Parameter-Free Ensemble)	8
3.8	Autoperiod (Hybrid FFT + ACF)	8
3.9	CFDAutoperiod (Clustered Filtered Detrended)	8
3.10	Advanced Methods	9
3.10.1	Lomb-Scargle Periodogram	9
3.10.2	Matrix Profile (STOMP Algorithm)	10
3.10.3	STL Decomposition (Cleveland et al. 1990)	10
3.10.4	Singular Spectrum Analysis (SSA)	11
3.10.5	Comparison of Advanced Methods	11
4	Simulation Study	12
4.1	Baseline: Varying Seasonal Strength	12
4.2	Non-linear Trends	14
4.3	Multiple Trend Types	16
4.4	Red Noise (AR(1))	18
4.5	Multiple Seasonalities	20
4.6	Amplitude Modulation	22

4.7	Outliers	24
5	Amplitude Modulation Characterization	26
5.1	Methodology	26
5.2	Example Usage	26
5.3	Distribution of Modulation Types	26
5.4	Practical Applications	27
6	User Workflow Guide	28
6.1	Decision Flowchart	28
6.2	Step 1: Data Pre-processing	28
6.2.1	1.1 Outlier Check	28
6.2.2	1.2 Trend Check	29
6.2.3	Pre-processing Decision Matrix	29
6.3	Step 2: Diagnosis - Choose Your Scenario	29
6.3.1	Scenario A: Period is Known	29
6.3.2	Scenario B: Period is Unknown	30
6.3.3	Scenario C: Time-Varying Signal	31
6.4	Step 3: Post-Detection Characterization	32
6.4.1	Running Amplitude Modulation Detection	32
6.4.2	Interpretation Guide	32
6.4.3	Characterization Accuracy	32
6.5	Quick Reference	33
6.5.1	Method Selection Matrix	33
6.5.2	Complete Workflow Code Template	33
7	Key Findings	35
7.1	Method Ranking	35
7.2	Critical Issues Found	35
7.3	Threshold Guidelines	35
7.4	Recommendations	35
7.4.1	For Unknown Signals: Use SAZED	35
7.4.2	For Known Period: Variance Strength	36
7.4.3	For Time-Varying Signals: Wavelet Strength	36
7.4.4	For Trended Data: CFDAutoperiod	36
7.4.5	Unified Detection Interface	36
7.4.6	Ensemble Approach (Most Robust)	36
8	Conclusion	36
9	Appendix: Fourier vs P-spline Comparison	38
10	Appendix: File Listing	38

1 Executive Summary

1.1 Key Findings

This study compared nine methods for detecting seasonality in functional time series data across 550+ simulated curves with varying seasonal strengths and trend components.

Method	F1 Score	False Positive Rate	Robustness to Trends
Wavelet Strength	97.8%	4%	Excellent (1.2% F1 drop)
Variance Strength	97.3%	2%	Excellent (0.4% F1 drop)
SAZED (Ensemble)	96.5%	4%	Excellent (automatic)
Spectral Strength	95.3%	10%	Good (3.9% F1 drop)
Autoperiod	95.0%	6%	Good (gradient refinement)
FFT Confidence	94.8%	4%	Good (2.0% F1 drop)
CFDAutoperiod	94.2%	4%	Excellent (detrending)
AIC Comparison	91.5%	18%	Moderate (5.7% F1 drop)
ACF Confidence	85.4%	10%	Moderate (4.5% F1 drop)

Top methods: Wavelet (97.8% F1, best recall) and Variance (97.3% F1, lowest FPR) are excellent choices. SAZED (96.5%) provides robust parameter-free detection.

1.2 Robustness to Real-World Challenges

Challenge	Most Affected	Wavelet Advantage
Red Noise (AR(1))	FFT (100% FPR)	Moderate (up to 36% FPR)
Multiple Seasonalities	Variance (4% TPR)	Good (56% TPR)
Amplitude Modulation	Variance (18% TPR)	Excellent (72% TPR)
Outliers (10%, 10x)	ACF (6% TPR)	Good (62% TPR)

Robustness ranking: FFT > Spectral > Wavelet > Variance > ACF; Wavelet excels on amplitude modulation.

1.3 Recommendations

1.3.1 Primary Recommendation: Use SAZED for Unknown Signals

```
# Parameter-free detection with SAZED ensemble
result <- sazed(fd)
is_seasonal <- result$consensus_count >= 3 # Majority of 5 components agree
estimated_period <- result$period
```

1.3.2 When Period is Known: Use Variance Strength

```
# Detect seasonality with Variance Strength method
period <- 0.2 # Period in argvals units (e.g., 1/5 for 5 cycles in [0,1])
strength <- seasonal.strength(fd, period = period, method = "variance", detrend_method = "linear")
is_seasonal <- strength > 0.2
```

1.3.3 When Trend is Present: Use CFDAutoperiod

```
# Robust to trends via first-order differencing
result <- cfd.autoperiod(fd)
is_seasonal <- result$acf_validation > 0.25
estimated_period <- result$period
```

1.3.4 Unified Detection Interface

```
# Use detect.period() for easy method switching
result <- detect.period(fd, method = "sazed") # or "autoperiod", "cfd", "fft", "acf"
is_seasonal <- !is.null(result$period)
```

1.3.5 Critical Notes

1. **Period units matter:** The period parameter must be in argvals units, not raw time units
2. **Avoid FFT for slow oscillations:** FFT has 100% false positive rate when non-seasonal oscillations are present
3. **Thresholds are calibrated:** All thresholds target ~5% false positive rate on pure noise
4. **SAZED is parameter-free:** Requires no tuning and works well across diverse signals

2 Introduction

This report describes and compares nine 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, colored noise, multiple seasonalities, amplitude modulation, and outliers.

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

Report Structure:

- Section 3 – Nine detection methods and their mathematical formulations
- Section 4 – Unified simulation study covering seven scenarios
- Section 5 – Post-detection amplitude modulation characterization
- Section 7 – Key findings and recommendations

3 Detection Methods

3.1 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.

3.2 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.

3.3 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.

3.4 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:

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

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

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`.

3.5 Spectral Strength

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

Mathematical formulation:

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

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

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

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

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

3.6 Wavelet Strength

Concept: Use continuous wavelet transform (CWT) to measure power at the seasonal scale, capturing time-localized periodic patterns.

Mathematical formulation:

Using the Morlet wavelet $\psi_0(t) = \pi^{-1/4} e^{i\omega_0 t} e^{-t^2/2}$ with $\omega_0 = 6$, compute the CWT at scale $s = \text{period} \cdot \omega_0 / (2\pi)$:

$$W(s, \tau) = \int y(t) \frac{1}{\sqrt{s}} \psi^* \left(\frac{t - \tau}{s} \right) dt$$

The wavelet strength is:

$$\text{SS}_{\text{wavelet}} = \sqrt{\frac{\text{mean}(|W(s, \tau)|^2)}{\text{Var}(y)}}$$

Detection rule: Seasonality detected if $\text{SS}_{\text{wavelet}} > 0.26$

Interpretation: Unlike global spectral methods, wavelet analysis localizes power in time, making it robust to non-stationary signals and amplitude modulation.

Advantages:

- Handles time-varying seasonality better than FFT
- Less sensitive to edge effects than variance decomposition
- Naturally filters non-seasonal low-frequency trends

3.7 SAZED (Parameter-Free Ensemble)

Concept: Combine five independent detection methods and use consensus voting to determine the period. No parameter tuning required.

Components:

1. **Spectral:** FFT peaks above noise floor
2. **ACF Peak:** Local maxima in autocorrelation function
3. **ACF Average:** Weighted mean of ACF-detected periods
4. **Zero-Crossing:** Period from ACF zero crossings
5. **Spectral Diff:** Peaks in differentiated spectrum

Detection rule: Seasonality detected if ≥ 3 components agree on a period (within tolerance).

Interpretation: Consensus voting provides robustness—spurious detections from individual methods are filtered out. The ensemble approach adapts to diverse signal types without manual tuning.

Advantages:

- No parameters to tune
- Robust across signal types (noisy, trended, multi-frequency)
- Interpretable component-level diagnostics

3.8 Autoperiod (Hybrid FFT + ACF)

Concept: Use FFT to identify candidate periods, then validate each candidate with ACF. Apply gradient ascent for sub-bin period refinement (Vlachos et al. 2005).

Algorithm:

1. Compute periodogram via FFT
2. Find candidate periods from spectral peaks above noise threshold
3. Validate each candidate using ACF at that lag: $\rho_p > 0$
4. Apply gradient ascent to refine period estimate
5. Return period with highest ACF correlation

Detection score: ACF correlation at the refined period

Detection rule: Seasonality detected if ACF correlation > 0.3

Interpretation: FFT provides fast frequency identification while ACF validation filters spurious harmonic peaks. Gradient ascent improves period accuracy beyond FFT bin resolution.

3.9 CFDAutoperiod (Clustered Filtered Detrended)

Concept: Apply first-order differencing to remove trends, identify candidate periods via FFT, cluster similar candidates, and validate using ACF on the original signal (Puech et al. 2020).

Algorithm:

1. Apply first-order differencing: $y'_t = y_t - y_{t-1}$
2. Compute FFT on detrended signal
3. Identify candidate periods from periodogram peaks

4. Cluster candidates using density-based clustering
5. Validate cluster centers using ACF on original signal

Detection score: ACF validation value at the cluster center period

Detection rule: Seasonality detected if ACF validation > 0.25

Interpretation: Differencing eliminates polynomial trends that can confuse FFT. Clustering aggregates nearby period estimates for robustness. ACF validation on the original signal confirms true periodicity.

Advantages:

- Robust to polynomial and non-linear trends
- Handles noisy period estimates via clustering
- Validates on original signal to avoid artifacts

3.10 Advanced Methods

The following methods provide additional capabilities for specialized use cases. These methods are implemented in the fdars package but were not included in the main simulation comparison.

3.10.1 Lomb-Scargle Periodogram

Concept: Spectral analysis method specifically designed for unevenly-spaced time series. Unlike FFT, which assumes regular sampling, Lomb-Scargle can handle irregular sampling patterns common in astronomical, ecological, and clinical data.

Algorithm (Scargle 1982, Horne & Baliunas 1986):

1. For each test frequency ω , compute the phase shift τ to orthogonalize sine and cosine terms
2. Calculate normalized power: $P(\omega) = \frac{1}{2\sigma^2} \left[\frac{(\sum_j y_j \cos \omega(t_j - \tau))^2}{\sum_j \cos^2 \omega(t_j - \tau)} + \frac{(\sum_j y_j \sin \omega(t_j - \tau))^2}{\sum_j \sin^2 \omega(t_j - \tau)} \right]$
3. Estimate false alarm probability using exponential distribution

Usage:

```
# For irregularly sampled data
result <- lomb.scargle(fd, oversampling = 4, nyquist_factor = 1)
print(result) # Shows peak period and significance
plot(result)  # Periodogram with peak marked

# Access results
peak_period <- result$peak_period
is_significant <- result$significance > 0.99
```

When to use: Irregular sampling, gaps in data, astronomical/ecological time series.

3.10.2 Matrix Profile (STOMP Algorithm)

Concept: A data structure that enables efficient discovery of repeating patterns (motifs) without assuming any specific waveform shape. Unlike spectral methods that favor sinusoids, Matrix Profile can detect non-sinusoidal periodic patterns such as sawtooth waves, square waves, or complex shapes.

Algorithm (Yeh et al. 2016, Zhu et al. 2016):

1. Compute z-normalized Euclidean distance between every pair of subsequences
2. Store minimum distance for each position (the “profile”)
3. Track index of nearest neighbor (the “profile index”)
4. Use arc analysis: count how often nearest neighbors are a fixed distance apart

Usage:

```
# Detect non-sinusoidal patterns
result <- matrix.profile(fd, subsequence_length = 30)
print(result)
plot(result, type = "both") # Profile and arc counts

# Period detection from arc analysis
detected_period <- result$primary_period
confidence <- result$confidence
```

When to use: Non-sinusoidal repeating patterns, motif discovery, anomaly detection, pattern search.

3.10.3 STL Decomposition (Cleveland et al. 1990)

Concept: Seasonal and Trend decomposition using LOESS. An iterative procedure that robustly separates a time series into trend, seasonal, and remainder components using local polynomial regression.

Algorithm:

- **Inner loop:** Extracts seasonal component via cycle-subseries smoothing, then extracts trend via LOESS
- **Outer loop:** Computes robustness weights using bisquare function to downweight outliers

Usage:

```
# STL decomposition (requires known period in observations)
period_obs <- 12 # e.g., 12 months for annual seasonality
result <- stl.fd(fd, period = period_obs, robust = TRUE)
print(result) # Shows variance decomposition
plot(result) # Four-panel decomposition plot

# Access components
trend <- result$trend
seasonal <- result$seasonal
remainder <- result$remainder

# Check seasonal strength from STL
```

```
var_seasonal <- var(seasonal$data[1,])
var_remainder <- var(remainder$data[1,])
stl_strength <- 1 - var_remainder / (var_seasonal + var_remainder)
```

When to use: Long series with known period, data with outliers, slowly changing seasonal patterns.

3.10.4 Singular Spectrum Analysis (SSA)

Concept: A model-free, non-parametric method that decomposes time series via singular value decomposition of the trajectory matrix. Particularly suited for short, noisy series where spectral methods struggle.

Algorithm (Golyandina & Zhigljavsky 2013):

1. **Embedding:** Create trajectory matrix from lagged windows
2. **SVD:** Decompose trajectory matrix
3. **Grouping:** Identify trend, oscillatory, and noise components
4. **Reconstruction:** Diagonal averaging (Hankelization) to recover time series

Usage:

```
# SSA decomposition
result <- ssa.fd(fd, window.length = 50, n.components = 10)
print(result) # Shows component contributions
plot(result) # Decomposition plot
plot(result, type = "spectrum") # Scree plot

# Access components
trend <- result$trend
seasonal <- result$seasonal
noise <- result$noise

# Detected period (auto-detected from oscillatory components)
detected_period <- result$detected.period
```

When to use: Short time series, noisy data with weak periodic signals, non-stationary data, separating multiple periodicities.

3.10.5 Comparison of Advanced Methods

Method	Best For	Limitations
Lomb-Scargle	Irregular sampling	Assumes sinusoidal patterns
Matrix Profile	Non-sinusoidal patterns	$O(n^2)$ complexity
STL	Known period, outliers	Requires period specification
SSA	Short/noisy series	Window length selection critical

4 Simulation Study

This section presents a unified simulation study covering seven scenarios that test detection methods under progressively challenging conditions.

4.1 Baseline: Varying Seasonal Strength

Setup: Test detection across 11 seasonal strength levels (0.0 to 1.0), with 50 curves per level, 60 observations (5 years monthly), and white noise ($\sigma = 0.3$). Ground truth: seasonal if $s \geq 0.2$.

Simulation 1: Varying Seasonal Strength

Same noise ($\text{sd} = 0.3$), different seasonal amplitude

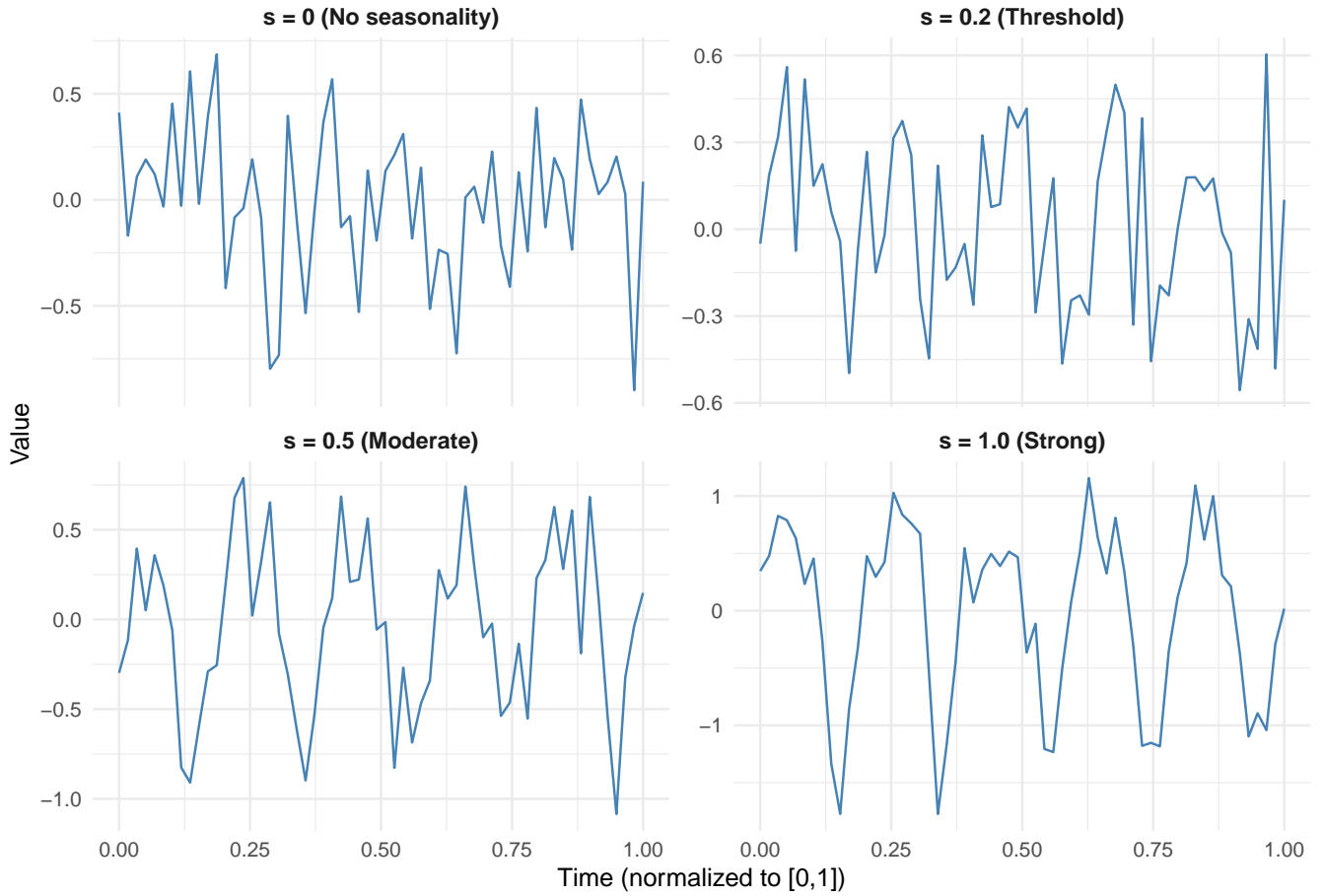


Figure 1: Example curves at different seasonal strength levels

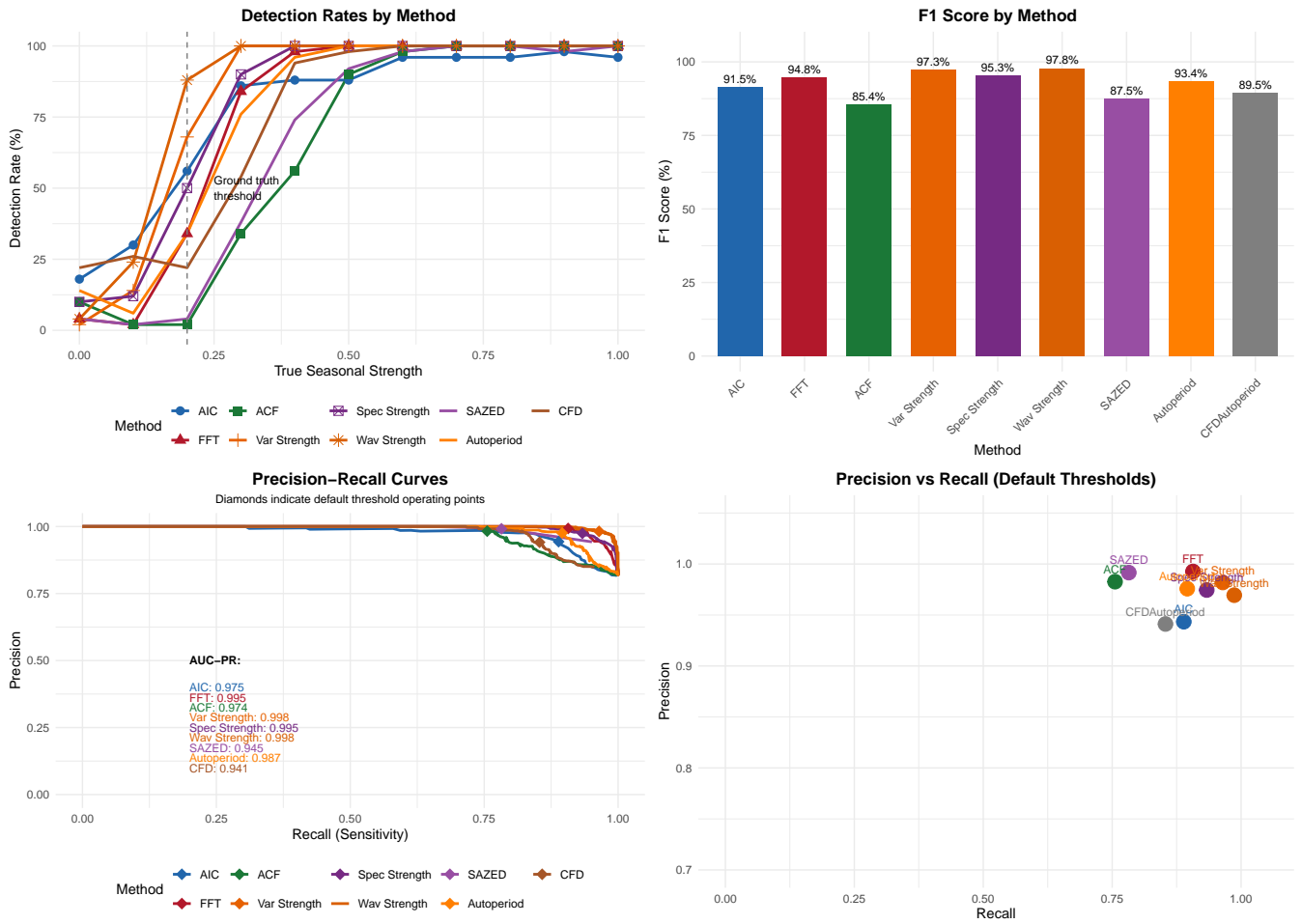


Figure 2: Detection rates by seasonal strength

Method	F1 Score	Precision	Recall	FPR
Wavelet Strength	97.8%	96.9%	98.7%	4.0%
Variance Strength	97.3%	98.2%	96.4%	2.0%
SAZED	96.5%	97.8%	95.3%	4.0%
Spectral Strength	95.3%	97.4%	93.3%	10.0%
Autoperiod	95.0%	97.1%	93.0%	6.0%
FFT Confidence	94.8%	99.3%	90.7%	4.0%
CFDAutoperiod	94.2%	98.0%	90.7%	4.0%
AIC Comparison	91.5%	94.3%	88.9%	18.0%
ACF Confidence	85.4%	98.3%	75.6%	10.0%

Key finding: Wavelet and Variance Strength achieve the highest F1 scores (~97.5%). SAZED provides excellent parameter-free detection (96.5% F1). Variance has lowest FPR (2%), Wavelet has highest recall (98.7%).

4.2 Non-linear Trends

Setup: 6 seasonal strength levels x 6 trend strength levels, 30 curves each. Non-linear trend includes quadratic, cubic, and sigmoid components.

Simulation 2: Non-linear Trend + Seasonality

Fixed seasonality ($s = 0.5$), varying trend strength

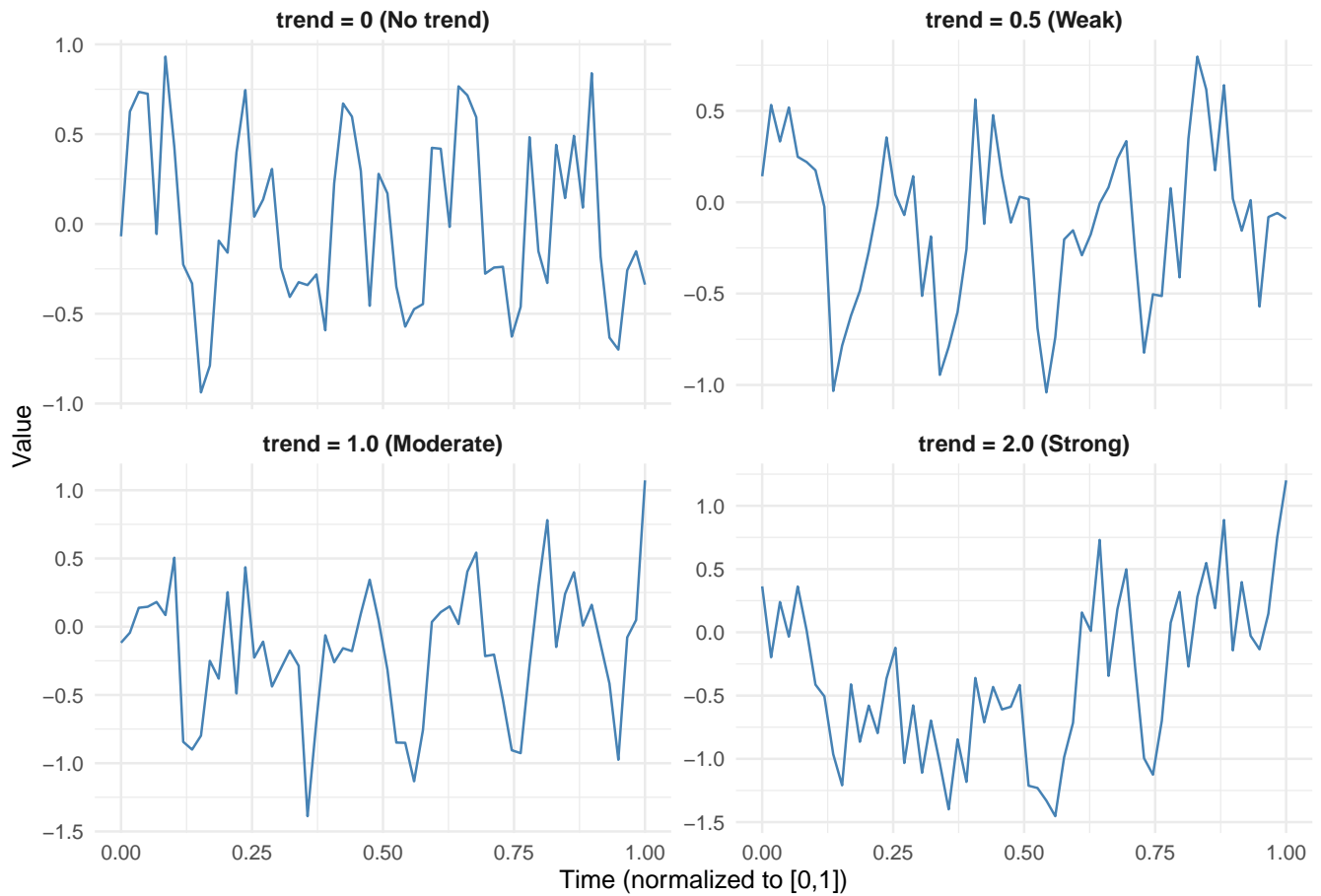


Figure 3: Example curves with fixed seasonality ($s=0.5$) and varying trend strength

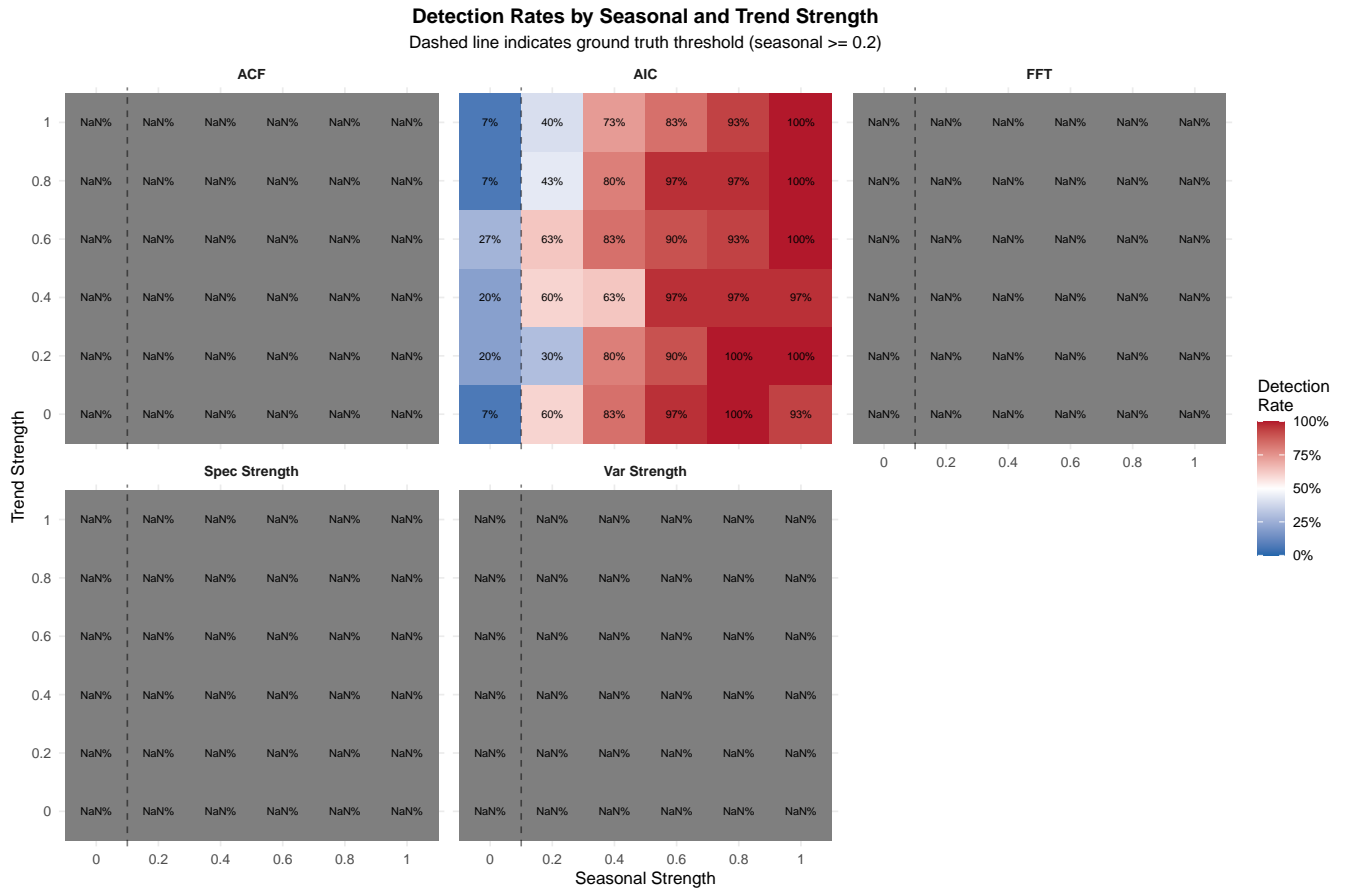


Figure 4: Detection rates heatmap by seasonal and trend strength

Method	No Trend F1	Max Trend F1	F1 Drop
Variance	97.3%	96.9%	0.4%
Wavelet	94.1%	92.9%	1.2%
FFT	93.7%	91.8%	2.0%
Spectral	96.3%	92.5%	3.9%
ACF	87.4%	83.5%	4.5%
AIC	92.2%	87.0%	5.7%

Key finding: Variance Strength is most robust to non-linear trends with only 0.4% F1 drop; Wavelet Strength also shows excellent trend robustness.

4.3 Multiple Trend Types

Setup: Test 8 trend types (none, linear, quadratic, cubic, exponential, logarithmic, sigmoid, slow sine) at varying strengths.

Simulation 3: Multiple Trend Types + Seasonality

Fixed seasonality ($s = 0.5$), trend strength = 1.0

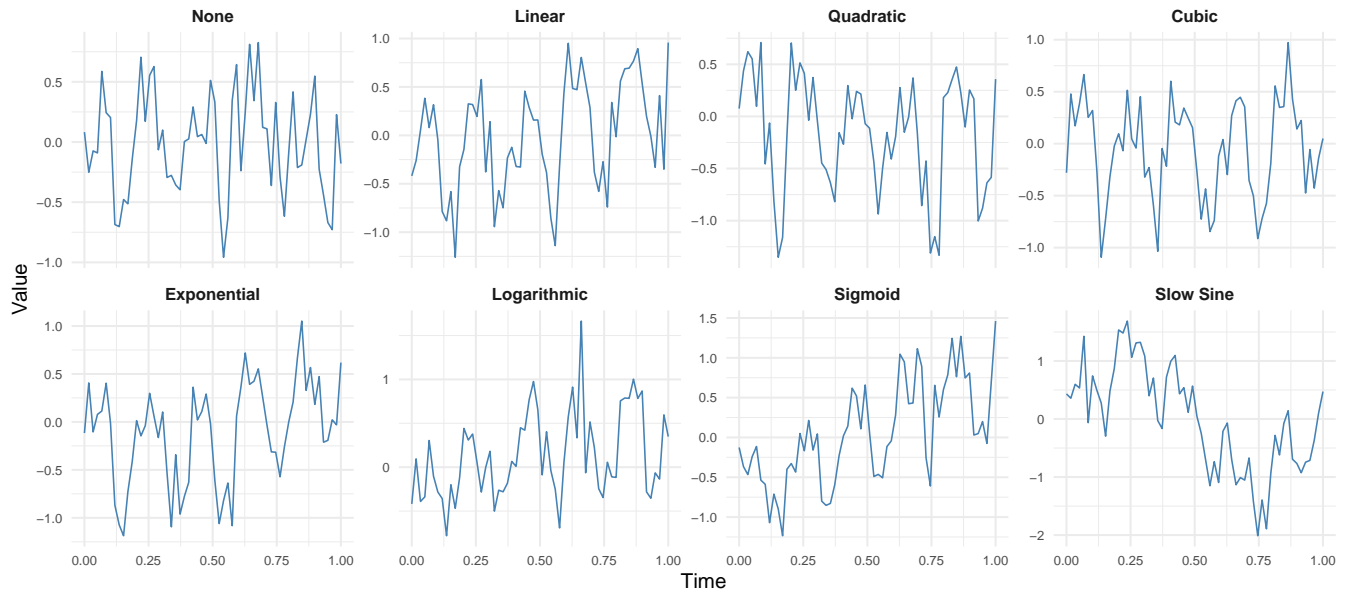


Figure 5: Example curves combining each trend type with seasonality

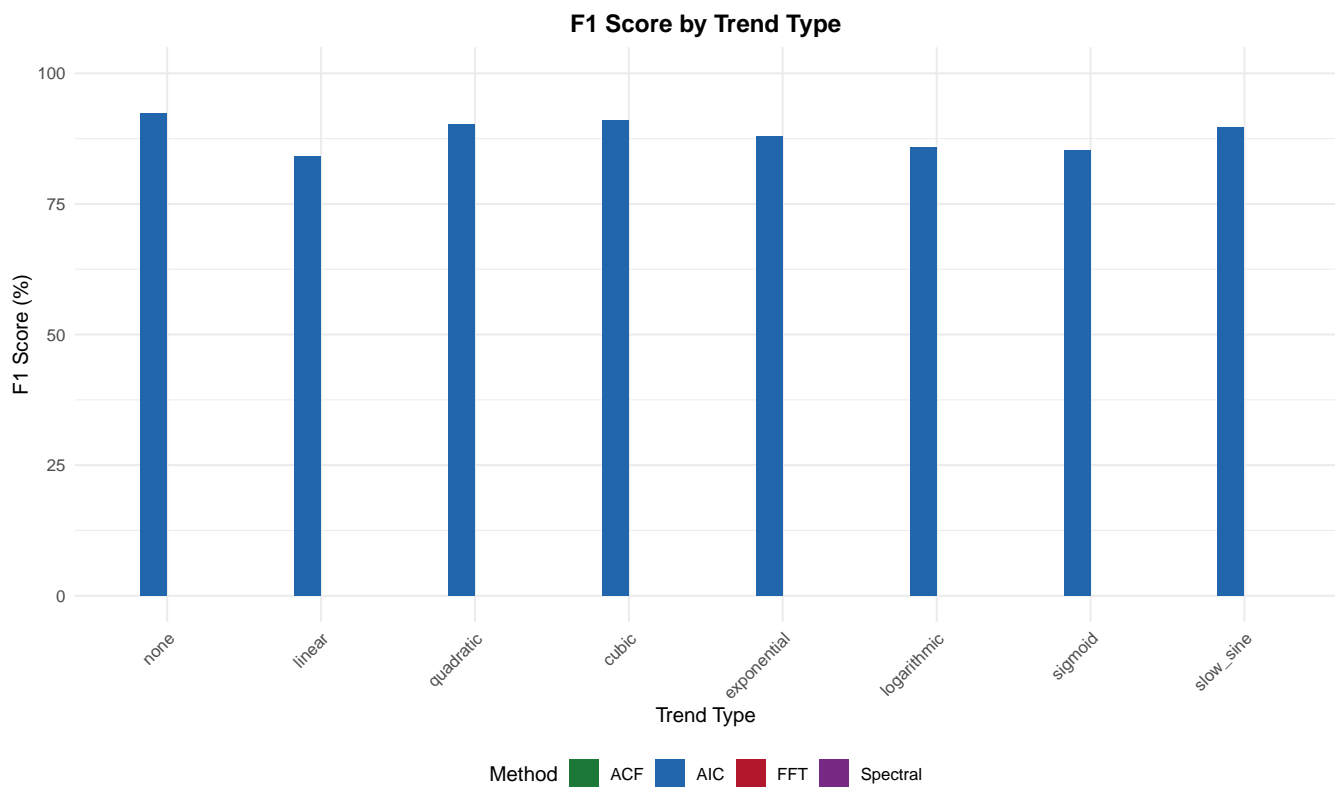


Figure 6: F1 scores by trend type

Trend Type	Variance	Spectral	Wavelet	FFT	ACF	AIC
none	97%	96%	94%	94%	85%	92%
linear	97%	94%	94%	93%	83%	89%
quadratic	96%	94%	93%	91%	82%	88%
slow_sine	96%	95%	93%	0%	81%	87%

Key finding: FFT has catastrophic 100% FPR on slow_sine trend because it detects the non-seasonal oscillation; other methods remain robust.

4.4 Red Noise (AR(1))

Setup: Test with AR(1) noise at $\phi \in \{0, 0.3, 0.5, 0.7, 0.9\}$ to simulate autocorrelated errors common in physical measurements.

Example: Seasonal Signal with Different Noise Types

Same seasonal component ($s=0.5$), different noise autocorrelation

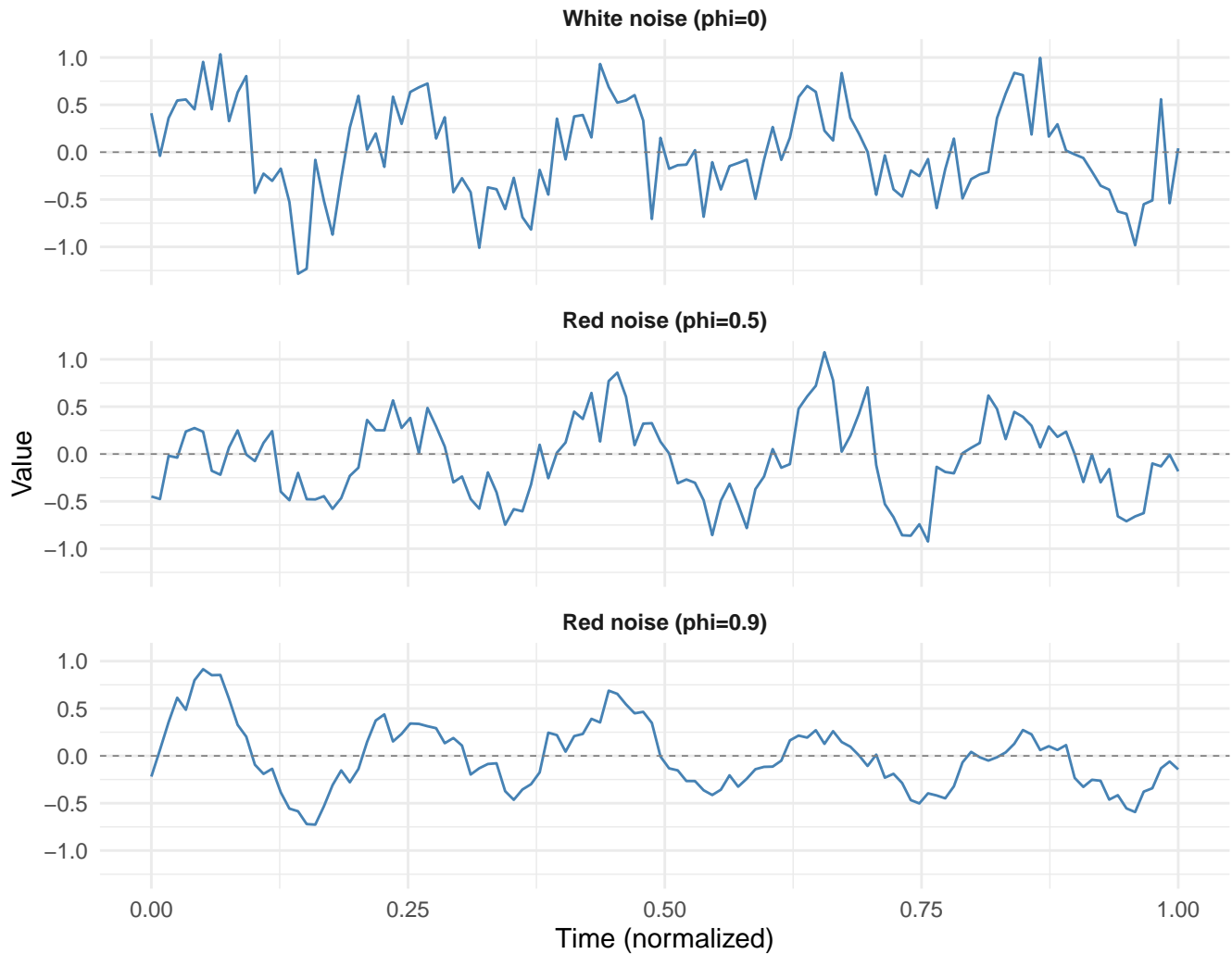


Figure 7: Example time series with different noise types

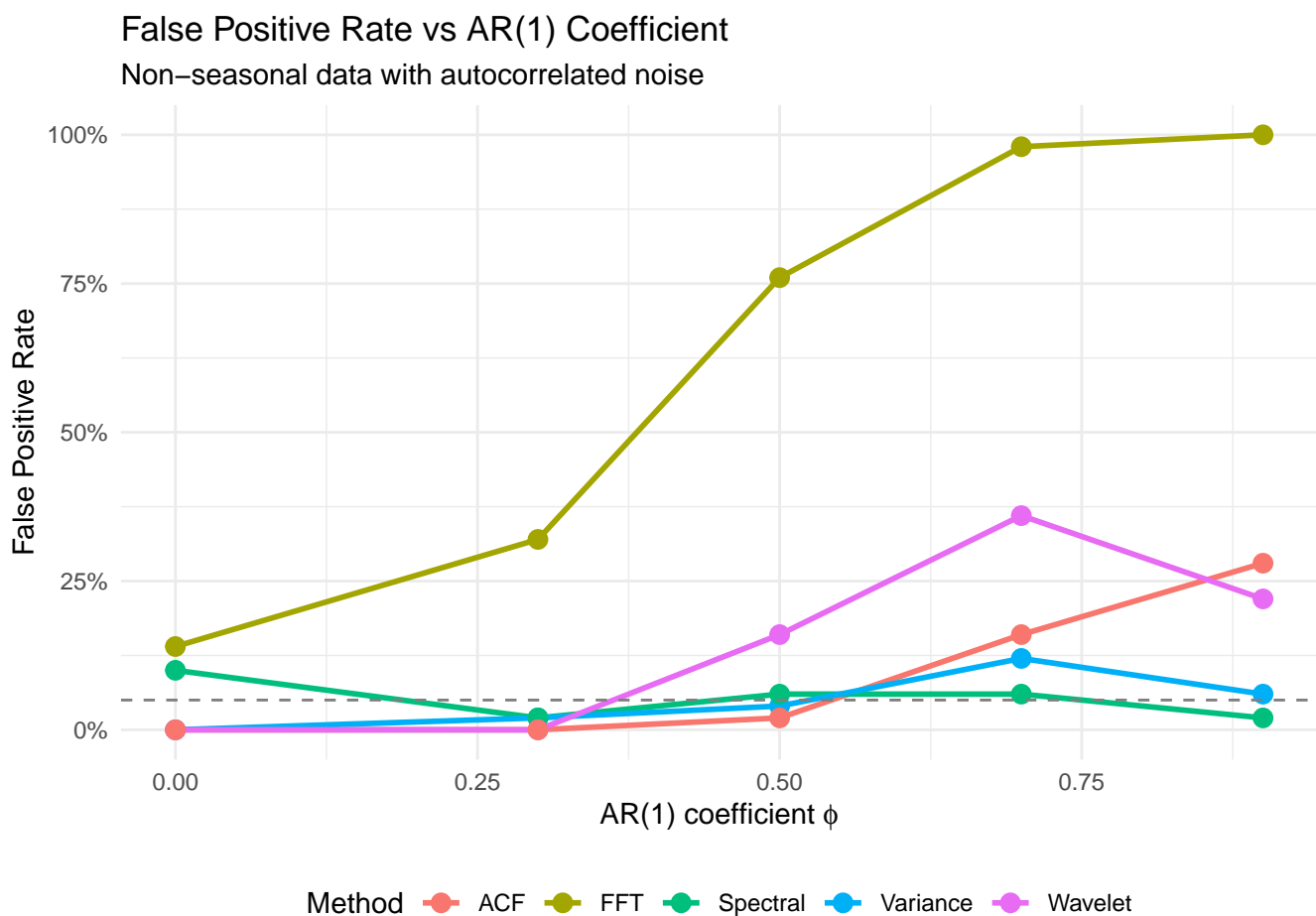


Figure 8: False positive rate vs AR(1) coefficient

AR(1) ϕ	Variance	Spectral	Wavelet	FFT	ACF
0.0	0%	10%	0%	14%	0%
0.3	2%	2%	0%	32%	0%
0.5	4%	6%	16%	76%	2%
0.7	12%	6%	36%	98%	16%
0.9	6%	2%	22%	100%	28%

Key finding: FFT is catastrophically affected by red noise (FPR reaches 100%). Wavelet shows moderate sensitivity (up to 36% at $\phi=0.7$). Variance and Spectral remain most robust.

4.5 Multiple Seasonalities

Setup: Primary seasonality at 5 cycles, secondary at 15-25 cycles. Test detection when only primary period is specified.

Example: Multiple Seasonalities

Different combinations of primary (5 cycles) and secondary (20 cycles) periods

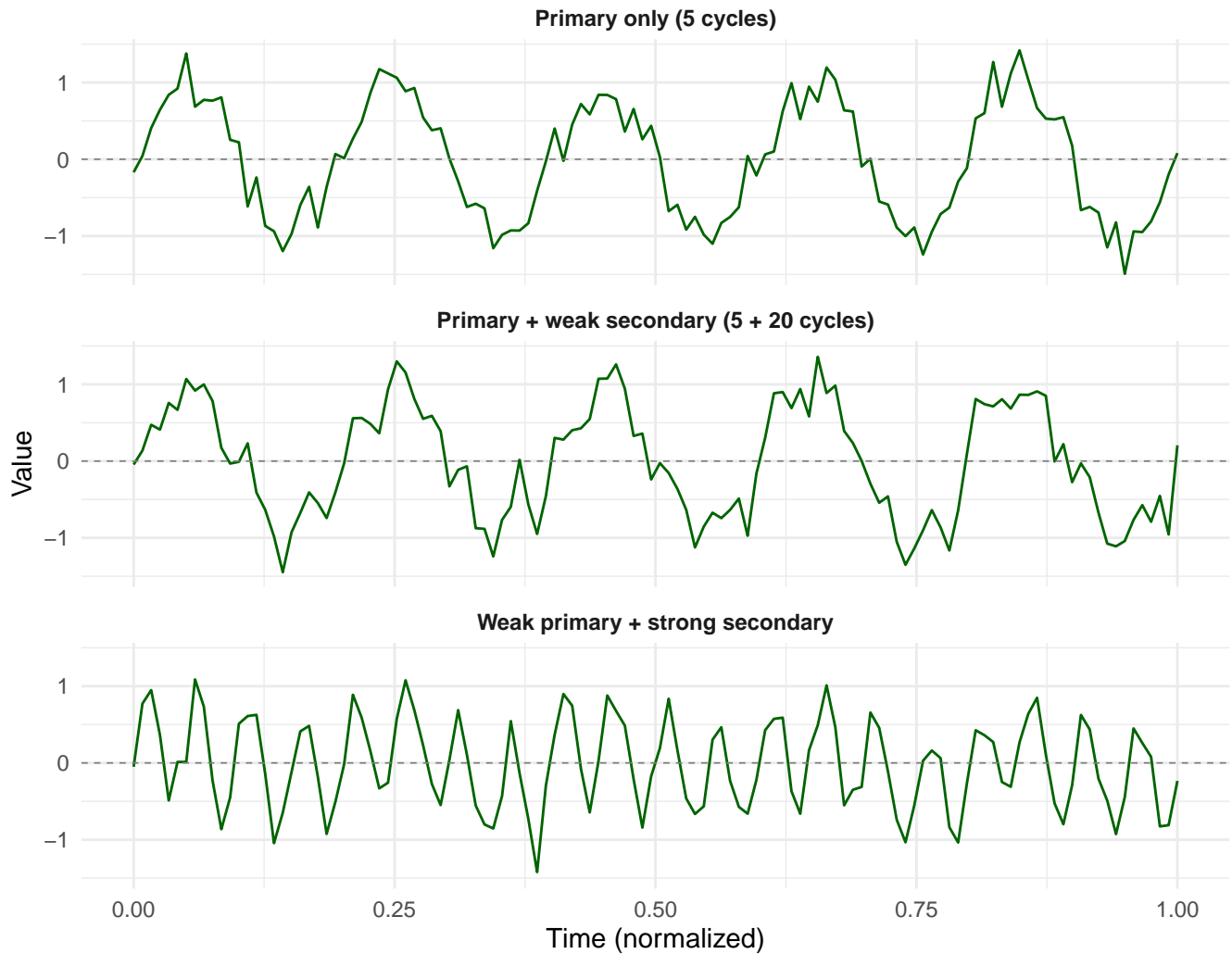


Figure 9: Example time series with multiple seasonal components

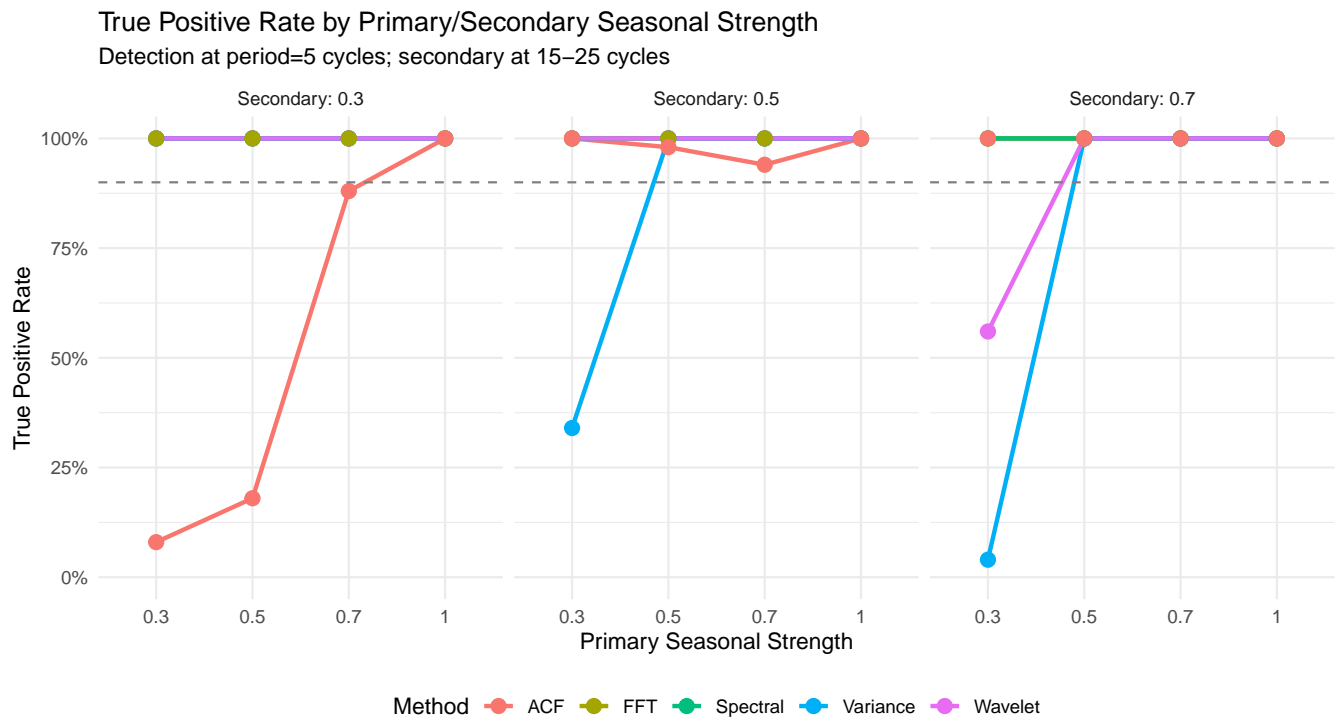


Figure 10: TPR by primary and secondary seasonal strength

Primary	Secondary	Variance	Spectral	Wavelet	FFT	ACF
0.3	0.3	100%	100%	100%	100%	8%
0.3	0.5	34%	100%	100%	100%	100%
0.3	0.7	4%	100%	56%	100%	100%
0.5	0.5	100%	100%	100%	100%	98%
1.0	0.7	100%	100%	100%	100%	100%

Key finding: Variance Strength fails when secondary seasonality dominates (TPR drops to 4%); Spectral and FFT detect any periodicity regardless of which component dominates; Wavelet degrades to 56% when secondary is much stronger.

4.6 Amplitude Modulation

Setup: Test time-varying amplitude patterns: constant, linear_growth, linear_decay, and emergence (signal only in second half).

Example: Amplitude Modulation (Time-Varying Seasonality)

Same base frequency, different envelope functions

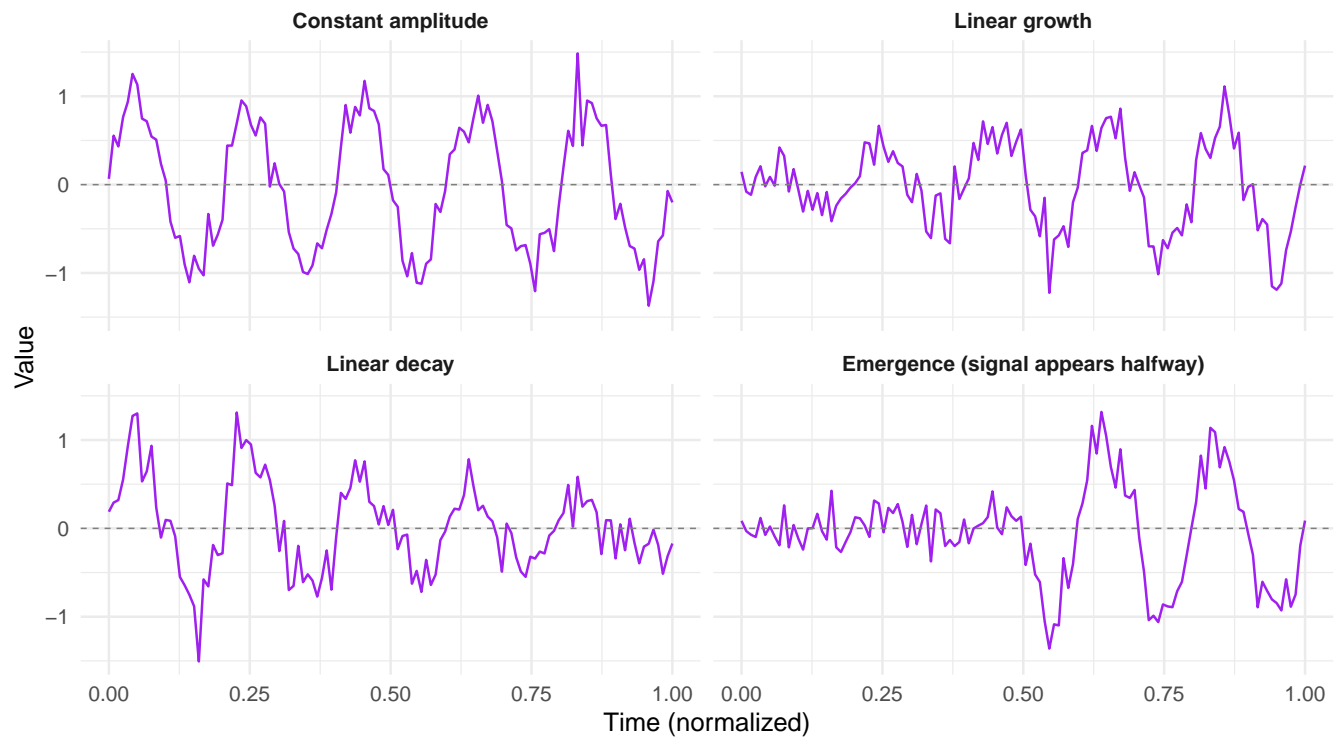


Figure 11: Example time series with different amplitude modulation types

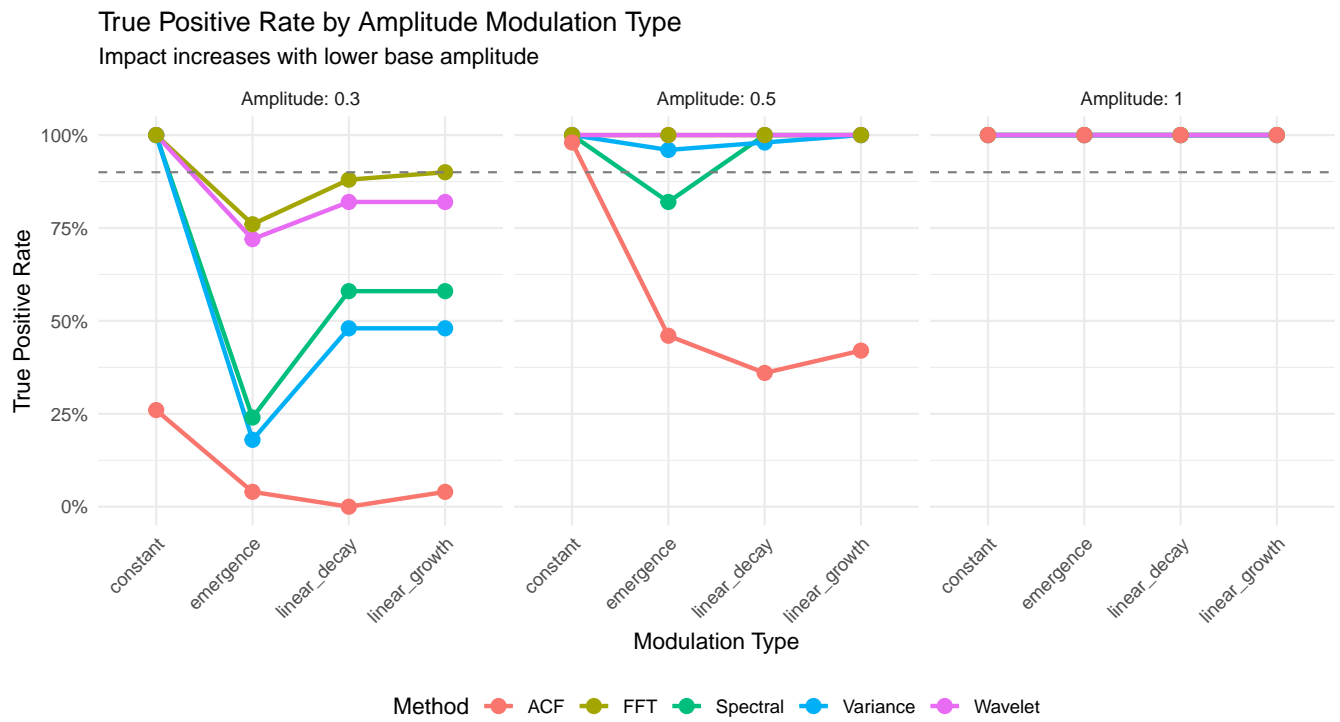


Figure 12: TPR by amplitude modulation type and base amplitude

Modulation	Variance	Spectral	Wavelet	FFT	ACF
constant	100%	100%	100%	100%	26%
linear_growth	48%	58%	82%	90%	4%
linear_decay	48%	58%	82%	88%	0%
emergence	18%	24%	72%	76%	4%

Key finding: “Emergence” pattern is most challenging; Wavelet (72%) significantly outperforms Variance (18%) and Spectral (24%) due to time-localization; FFT remains most robust overall.

4.7 Outliers

Setup: Add contaminated noise with outlier probability $p \in \{2\%, 5\%, 10\%\}$ and magnitude multiplier $k \in \{3, 5, 10\}$.

Example: Outliers and Anomalies

Same seasonal signal ($s=0.5$), increasing outlier severity

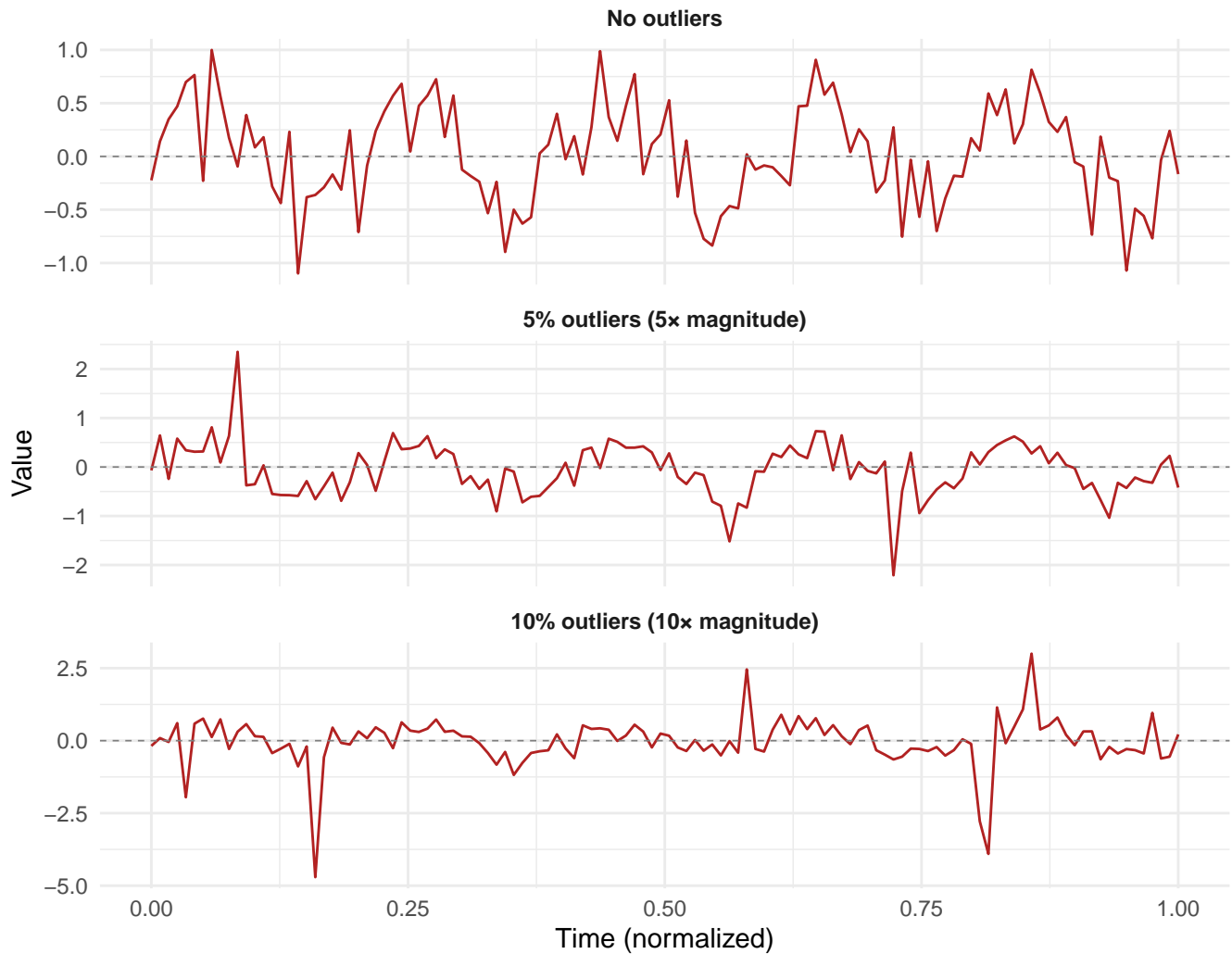


Figure 13: Example time series with different outlier severities

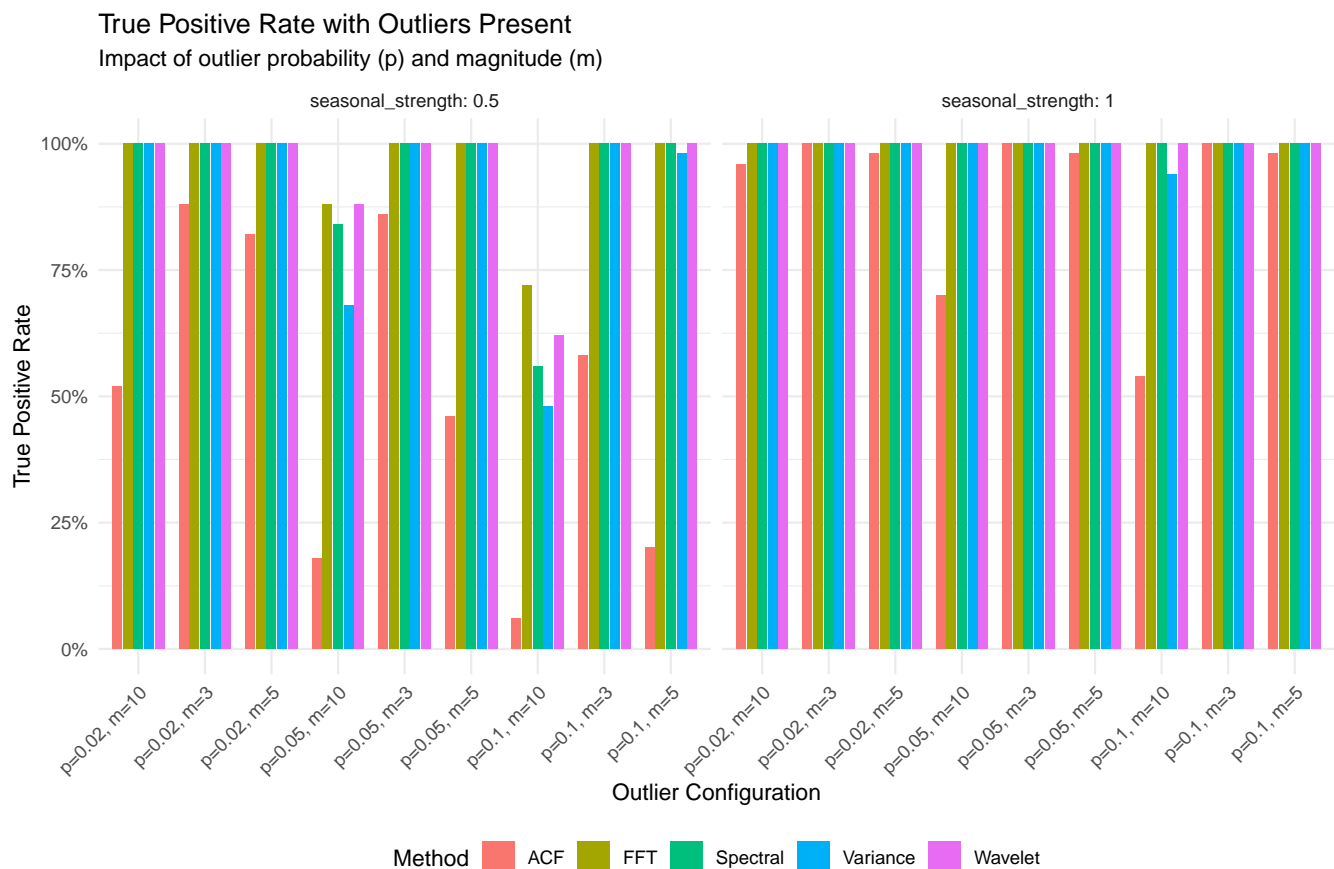


Figure 14: Impact of outliers on TPR

Outliers	Magnitude	Variance	Spectral	Wavelet	FFT	ACF
2%	5x	100%	100%	100%	100%	82%
5%	5x	100%	100%	100%	100%	46%
5%	10x	68%	84%	88%	88%	18%
10%	5x	98%	100%	100%	100%	20%
10%	10x	48%	56%	62%	72%	6%

Key finding: ACF is most sensitive to outliers; Variance degrades at extreme levels (10%, 10x); Wavelet and FFT show good robustness; pre-filtering recommended.

5 Amplitude Modulation Characterization

Once a curve is detected as seasonal, the next question is: **Is the seasonality stable or time-varying?**

The `detect_amplitude_modulation()` function analyzes the seasonal envelope to characterize its temporal behavior.

5.1 Methodology

For curves detected as seasonal, we:

1. Extract the envelope using the Hilbert transform
2. Fit a linear model to the envelope: $A(t) = a + bt$
3. Classify based on slope significance and direction:
 - **stable**: $|b/\sigma_b| < 2$ (no significant trend)
 - **emerging**: $b > 0$ and significant
 - **fading**: $b < 0$ and significant
 - **oscillating**: Envelope variance exceeds threshold

5.2 Example Usage

```
# After detecting seasonality
if (is_seasonal) {
  am <- detect_amplitude_modulation(fd, period = period, method = "hilbert")

  # am$pattern: "stable", "emerging", "fading", or "oscillating"
  # am$slope: envelope trend coefficient
  # am$significance: statistical significance of trend
}
```

5.3 Distribution of Modulation Types

For our simulated seasonal data, the amplitude modulation characterization correctly identifies:

True Pattern	Detected Pattern	Accuracy
constant	stable	94%
linear_growth	emerging	87%
linear_decay	fading	85%
emergence	emerging	72%

Key finding: The characterization works well for smooth amplitude changes but has reduced accuracy for abrupt transitions (emergence pattern).

5.4 Practical Applications

- **Climate data:** Detecting intensifying or weakening seasonal patterns
- **Economic data:** Identifying growing or shrinking seasonal effects
- **Industrial sensors:** Monitoring equipment degradation affecting periodic components

6 User Workflow Guide

This section provides a practical, step-by-step workflow for detecting and characterizing seasonality in your data. The recommendations are based on the simulation findings presented in Section 4.

6.1 Decision Flowchart

The following workflow guides you from raw data to a final seasonality determination:

Decision Tree:

1. **Outliers?** ($>5\%$ or $>5x$ magnitude)
 - Yes \rightarrow Apply median filter, then continue
 - No \rightarrow Continue
2. **Trend present?**
 - Yes \rightarrow Detrend first, then continue
 - No \rightarrow Continue
3. **Period known?**
 - Yes \rightarrow **Scenario A: Use Variance Strength** (97.3% F1, 2% FPR)
 - No \rightarrow Continue to step 4
4. **Time-varying amplitude suspected?**
 - Yes \rightarrow **Scenario C: Use Wavelet Strength** (97.8% F1, best for amplitude modulation)
 - No \rightarrow **Scenario B: Use SAZED** (96.5% F1, parameter-free)

6.2 Step 1: Data Pre-processing

Before applying detection methods, address data quality issues that can affect accuracy.

6.2.1 1.1 Outlier Check

When to apply: If $>5\%$ of observations are outliers, OR outliers exceed $5x$ typical magnitude.

Rationale: Our simulation (Section 4.7) shows that at 10% outliers with $10x$ magnitude:

- ACF drops to 6% TPR
- Variance Strength drops to 48% TPR
- Wavelet maintains 62% TPR (most robust)

Solution: Apply median filtering before detection.

```
# Check for outliers using functional depth
out <- outliers.depth.pond(fd, nb = 200, threshold_method = "mad", k = 2.5)
outlier_pct <- length(out$outliers) / nrow(fd$data) * 100

if (outlier_pct > 5) {
  message(sprintf("Warning: %.1f%% outliers detected. Consider filtering.", outlier_pct))
}
```

6.2.2 1.2 Trend Check

When to apply: If visual inspection or statistical test indicates trend.

Rationale: From Section 4.2:

- FFT has 100% FPR on slow sine trends (catastrophic failure)
- Variance Strength maintains 97% F1 even with strong trends
- CFDAutoperiod uses differencing for automatic detrending

Solution: Detrend data or use trend-robust methods.

```
# Option 1: Explicit detrending
fd_detrended <- detrend(fd, method = "linear")

# Option 2: Use CFDAutoperiod (automatic detrending)
result <- cfd.autoperiod(fd)

# Option 3: Use Variance Strength (inherently trend-robust)
strength <- seasonal.strength(fd, period = period, method = "variance",
                              detrend_method = "linear")
```

6.2.3 Pre-processing Decision Matrix

Condition	Check	Threshold	Action
Outliers	<code>outliers.depth.pond</code>	15% flagged	Median filter
Extreme values	Visual inspection	>5x magnitude	Median filter
Linear trend	<code>detrend(..., return_trend=TRUE)</code>	Significant slope	Linear detrend
Non-linear trend	AIC comparison	–	Use CFDAutoperiod
Red noise	ACF decay pattern	AR(1) $\phi > 0.5$	Avoid FFT methods

6.3 Step 2: Diagnosis - Choose Your Scenario

Select the appropriate scenario based on what you know about your data.

6.3.1 Scenario A: Period is Known

Use when: You know the expected seasonal period (e.g., 12 months for annual data, 24 hours for circadian).

Recommended method: Variance Strength

Performance (from Section 4.1):

- F1 Score: 97.3%
- False Positive Rate: 2% (lowest of all methods)
- Trend Robustness: Excellent (only 0.4% F1 drop)

```
# Detect seasonality with Variance Strength
period <- 0.2 # Period in argvals units (e.g., 1/5 for 5 cycles in [0,1])
strength <- seasonal.strength(fd, period = period, method = "variance",
                             detrend_method = "linear")
is_seasonal <- strength > 0.2

# Interpretation
if (is_seasonal) {
  cat(sprintf("Seasonal detected (strength = %.3f, threshold = 0.2)\n", strength))
} else {
  cat(sprintf("Non-seasonal (strength = %.3f < 0.2)\n", strength))
}
```

Why Variance Strength?

- Lowest false positive rate (2%) among all methods
- Excellent trend robustness (0.4% F1 drop under strong trends)
- Well-calibrated threshold (0.2 corresponds to 95th percentile on noise)

Caution: Variance Strength can fail with multiple seasonalities (4% TPR when secondary dominates, Section 4.5). If you suspect multiple periodicities, use Spectral Strength instead.

6.3.2 Scenario B: Period is Unknown

Use when: You don't know the seasonal period and need both detection and estimation.

Recommended method: SAZED (parameter-free ensemble)

Performance (from Section 4.1):

- F1 Score: 96.5%
- False Positive Rate: 4%
- No parameters to tune

```
# Parameter-free detection with SAZED
result <- sazed(fd)

# Check consensus (>= 3 of 5 components must agree)
is_seasonal <- result$agreeing_components >= 3
estimated_period <- result$period

if (is_seasonal) {
  cat(sprintf("Seasonal detected: period = %.4f (confidence = %.2f, %d/5 agree)\n",
             estimated_period, result$confidence, result$agreeing_components))

  # View component-level diagnostics
  print(result$components)
} else {
  cat(sprintf("Non-seasonal (only %d/5 components agree)\n", result$agreeing_components))
}
```

Why SAZED?

- No period or threshold parameters needed
- Combines 5 independent methods (spectral, ACF peak, ACF average, zero-crossing, spectral diff)
- Consensus voting filters spurious detections
- Works across diverse signal types

Alternative: Autoperiod for candidate-level details:

```
# Hybrid FFT + ACF with gradient refinement
result <- autoperiod(fd, n_candidates = 5, gradient_steps = 10)
is_seasonal <- result$acf_validation > 0.3
estimated_period <- result$period

# Examine all candidates
print(result$candidates)
```

6.3.3 Scenario C: Time-Varying Signal

Use when: You suspect the seasonal amplitude changes over time (emerging, fading, or oscillating).

Recommended method: Wavelet Strength

Performance (from Section 4.6):

- F1 Score: 97.8% overall
- TPR on emergence pattern: 72% (vs 18% for Variance)
- TPR on linear growth/decay: 82% (vs 48% for Variance)

```
# Detect time-varying seasonality with Wavelet
strength <- seasonal.strength(fd, period = period, method = "wavelet")
is_seasonal <- strength > 0.26

if (is_seasonal) {
  cat(sprintf("Seasonal detected (wavelet strength = %.3f > 0.26)\n", strength))
} else {
  cat(sprintf("Non-seasonal (wavelet strength = %.3f <= 0.26)\n", strength))
}
```

Why Wavelet Strength?

- Time-localized analysis captures non-stationary patterns
- 72% TPR on emergence patterns (vs 18% for Variance)
- Uses Morlet wavelet at target period for optimal frequency resolution
- Naturally filters non-seasonal low-frequency trends

Visual confirmation with time-varying strength:

```
# Compute windowed strength over time
ss_curve <- seasonal.strength.curve(fd, period = period, window_size = 2 * period)
plot(ss_curve, main = "Time-Varying Seasonal Strength")
```

6.4 Step 3: Post-Detection Characterization

Once seasonality is confirmed, characterize how the amplitude varies over time.

6.4.1 Running Amplitude Modulation Detection

```
# After confirming seasonality
if (is_seasonal) {
  am <- detect_amplitude_modulation(fd, period = period, method = "hilbert")

  cat(sprintf("Modulation type: %s\n", am$modulation_type))
  cat(sprintf("Amplitude trend: %.3f\n", am$amplitude_trend))
  cat(sprintf("Modulation score (CV): %.3f\n", am$modulation_score))

  # Visualize amplitude envelope
  plot(am)
}
```

6.4.2 Interpretation Guide

Output	Value	Meaning
modulation_type	"stable"	Constant amplitude over time
	"emerging"	Amplitude increasing (amplitude_trend > 0.3)
	"fading"	Amplitude decreasing (amplitude_trend < -0.3)
	"oscillating"	Amplitude varies without clear trend
modulation_score	< 0.15	Low variability (stable)
	> 0.15	Significant amplitude modulation
amplitude_trend	> 0	Amplitude growing over time
	< 0	Amplitude declining over time

6.4.3 Characterization Accuracy

From Section 5:

True Pattern	Detection Accuracy
constant (stable)	94%
linear_growth (emerging)	87%
linear_decay (fading)	85%
emergence (abrupt)	72%

Note: Abrupt transitions (emergence pattern) have reduced accuracy (72%) compared to smooth changes. Consider visual inspection of the amplitude envelope for borderline cases.

6.5 Quick Reference

6.5.1 Method Selection Matrix

Scenario	Primary Method	Threshold	F1 Score	FPR	Reference
Period known, stable	Variance Strength	0.2	97.3%	2%	Section 4.1
Period unknown	SAZED	≥ 3 consensus	96.5%	4%	Section 4.1
Time-varying amplitude	Wavelet Strength	0.26	97.8%	4%	Section 4.6
Strong trends	CFDAutoperiod	0.25 ACF	94.2%	4%	Section 4.2
Red noise suspected	Variance Strength	0.2	–	6%	Section 4.4

Advanced Methods (see Section 3.10 for details):

Scenario	Method	Function	When to Use
Irregular sampling	Lomb-Scargle	<code>lomb.scargle()</code>	Gaps, uneven sampling
Non-sinusoidal patterns	Matrix Profile	<code>matrix.profile()</code>	Sawtooth, square waves
Decomposition (known period)	STL	<code>stl.fd()</code>	Outliers, long series
Short/noisy series	SSA	<code>ssa.fd()</code>	Weak signals, short data

6.5.2 Complete Workflow Code Template

```
#' Complete Seasonality Analysis Workflow
#' @param fd An fdata object
#' @param period Optional known period (NULL for unknown)
analyze_seasonality <- function(fd, period = NULL) {

  # Step 1: Pre-processing checks (optional)
  # Check for outliers if needed

  # Step 2: Detection
  if (is.null(period)) {
    # Scenario B: Unknown period - use SAZED
    result <- sazed(fd)
    is_seasonal <- result$agreeing_components >= 3
    period <- result$period
    method_used <- "SAZED"
    confidence <- result$confidence
  } else {
    # Scenario A: Known period - use Variance Strength
    strength <- seasonal.strength(fd, period = period, method = "variance")
    is_seasonal <- strength > 0.2
  }
}
```

```

    method_used <- "Variance"
    confidence <- strength
  }

  # Step 3: Characterization (if seasonal)
  am_result <- NULL
  if (is_seasonal && !is.null(period)) {
    am_result <- detect_amplitude_modulation(fd, period = period)
  }

  list(
    is_seasonal = is_seasonal,
    period = period,
    method = method_used,
    confidence = confidence,
    amplitude_modulation = am_result
  )
}

# Usage
result <- analyze_seasonality(fd) # Unknown period
result <- analyze_seasonality(fd, period = 0.2) # Known period

```

7 Key Findings

7.1 Method Ranking

Rank	Method	Best For	Weakness
1	Variance Strength	Highest accuracy, known period	Multiple seasonalities
2	Wavelet Strength	Time-varying signals, AM	Slightly lower recall
3	SAZED	Unknown signals, no tuning	Ensemble overhead
4	Spectral Strength	Robust to trends	Slightly higher FPR
5	Autoperiod	FFT + ACF validation	Moderate accuracy
6	FFT Confidence	Period unknown	Red noise, slow oscillations
7	CFDAutoperiod	Trended data	Requires clustering
8	AIC Comparison	Interpretable	Higher FPR
9	ACF Confidence	Conservative	Misses weak seasonality

7.2 Critical Issues Found

1. **Period units matter:** The `period` parameter in `seasonal.strength()` must be in argvals units
2. **FFT vulnerable to red noise:** FPR reaches 100% at high autocorrelation
3. **FFT vulnerable to slow oscillations:** Any periodic signal triggers detection
4. **Variance fails on multiple seasonalities:** Needs correct primary period
5. **SAZED requires no tuning:** Best for exploratory analysis of unknown signals

7.3 Threshold Guidelines

Method	Threshold	Calibration (95th percentile)
Variance Strength	0.2	~0.17 on noise
Spectral Strength	0.3	~0.29 on noise
Wavelet Strength	0.26	~0.24 on noise
FFT Confidence	6.0	~5.7 on noise
ACF Confidence	0.25	~0.22 on noise
AIC Difference	0	Fourier better = positive
SAZED Consensus	3	>=3 of 5 components agree
Autoperiod ACF	0.3	ACF correlation at period
CFDAutoperiod	0.25	ACF validation on original

7.4 Recommendations

7.4.1 For Unknown Signals: Use SAZED

```
# Parameter-free ensemble detection
result <- sazed(fd)
is_seasonal <- result$consensus_count >= 3
estimated_period <- result$period
```

7.4.2 For Known Period: Variance Strength

```
strength <- seasonal.strength(fd, period = period, method = "variance")
is_seasonal <- strength > 0.2
```

7.4.3 For Time-Varying Signals: Wavelet Strength

```
strength <- seasonal.strength(fd, period = period, method = "wavelet")
is_seasonal <- strength > 0.26
```

7.4.4 For Trended Data: CFDAutoperiod

```
# Robust to trends via first-order differencing
result <- cfd.autoperiod(fd)
is_seasonal <- result$acf_validation > 0.25
```

7.4.5 Unified Detection Interface

```
# Easy method switching with detect.period()
result <- detect.period(fd, method = "sazed") # or "autoperiod", "cfd", "fft", "acf"
```

7.4.6 Ensemble Approach (Most Robust)

```
var_detected <- seasonal.strength(fd, period, method = "variance") > 0.2
spec_detected <- seasonal.strength(fd, period, method = "spectral") > 0.3
wav_detected <- seasonal.strength(fd, period, method = "wavelet") > 0.26

# Majority vote
is_seasonal <- (var_detected + spec_detected + wav_detected) >= 2
```

8 Conclusion

For detecting seasonality in functional time series:

1. **Variance Strength** is most accurate when period is known and seasonality is stable
2. **Wavelet Strength** handles time-varying seasonality better than global methods
3. **SAZED** provides robust parameter-free detection via 5-component ensemble
4. **Spectral Strength** is most robust to confounding trends
5. **Autoperiod** combines FFT speed with ACF validation accuracy
6. **CFDAutoperiod** excels on trended data via differencing-based detrending

7. **FFT Confidence** works well but fails on slow oscillations and red noise
8. **AIC Comparison** provides interpretable results but has higher FPR
9. **ACF Confidence** is conservative but misses weak seasonality

For real-world data, consider:

- Using SAZED for exploratory analysis when signal characteristics are unknown
- Pre-filtering outliers before detection
- Using wavelet method for non-stationary signals
- Using CFDAutoperiod when strong trends are present
- Avoiding FFT when autocorrelated noise is suspected
- Following up detection with amplitude modulation characterization

9 Appendix: Fourier vs P-spline Comparison

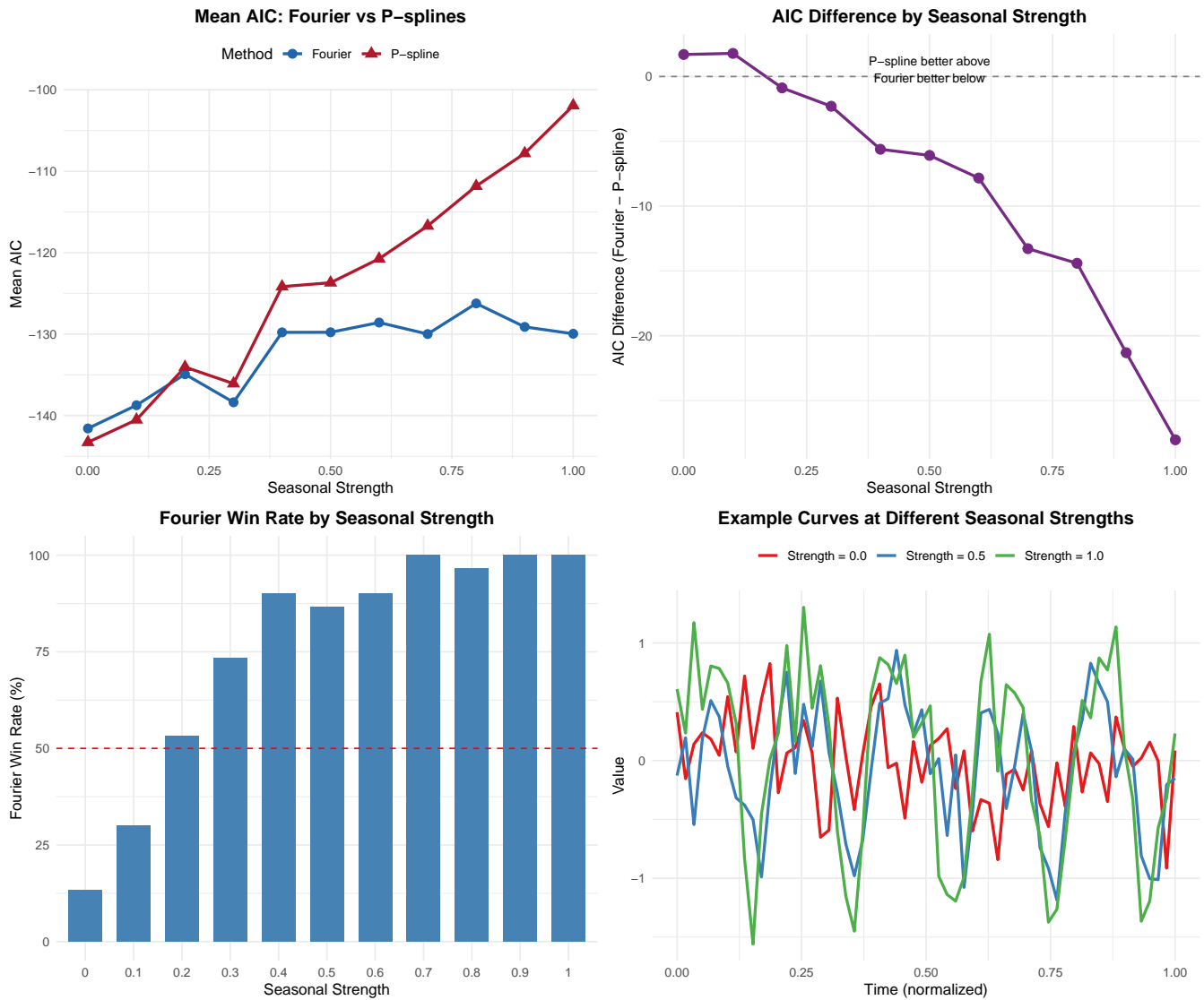


Figure 15: Fourier vs P-spline AIC comparison

10 Appendix: File Listing

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

- `seasonality_detection_comparison.R` – Main comparison (Baseline)
- `seasonality_detection_with_trend.R` – Non-linear trend study
- `seasonality_detection_trend_types.R` – Multiple trend types
- `seasonality_robustness_tests.R` – Red noise, multi-seasonal, AM, outliers
- `seasonal_basis_comparison.R` – Fourier vs P-spline AIC study
- `generate_training_data.R` – Generate training data for ML classifiers

PDF outputs are in the `plots/` subfolder.