Seasonality Detection Methods: A Comparative Study

fdars Package

2026-01-01

Table of contents

1 Executive Summary

1.1 Key Findings

This study compared five methods for detecting seasonality in functional time series data across 550+ simulated curves with varying seasonal strengths and trend components. **Note**: Results are based on idealized conditions (white noise, single seasonality); see Section 5 for real-world challenges.

Method	F1 Score	False Positive Rate	Robustness to Trends
Variance Strength	**97.3%**	**2%**	Excellent (0.4% F1 drop)
Spectral Strength	95.3%	10%	Good (3.9% F1 drop)
FFT Confidence	94.8%	4%	Good (2.0% F1 drop)
AIC Comparison	91.5%	18%	Moderate (5.7% F1 drop)
ACF Confidence	85.4%	10%	Moderate (4.5% F1 drop)

Winner: Variance Strength achieves the highest accuracy with the lowest false positive rate and is most robust to non-linear trends.

1.2 Robustness to Real-World Challenges

We tested all methods against challenging real-world scenarios:

Challenge	Most Affected	Key Result
Red Noise (AR(1))	FFT	FPR reaches 100% at ϕ=0.9
Multiple Seasonalities	Variance	TPR drops to 4% when secondary dominates
Amplitude Modulation	Variance, ACF	TPR drops to 18% for "emergence" pattern
Outliers (10%, 10×)	ACF	TPR drops to 2%; Variance to 40%

Robustness ranking: FFT > Spectral > Variance > ACF (FFT most robust overall, but vulnerable to red noise)

1.3 Recommendations

1.3.1 Primary Recommendation: Use Variance Strength

```r
# 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 = "linear")
is_seasonal <- strength > 0.2
```

1.3.2 When Period is Unknown: Two-Step Approach

```r
# Step 1: Estimate period using FFT (no period required)
result <- estimate_period(fd, method = "fft", detrend = "linear")
estimated_period <- result$period

# Step 2: Measure strength with estimated period
strength <- seasonal_strength(fd, period = estimated_period, method = "variance")
is_seasonal <- strength > 0.2
```

1.3.3 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

2 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?**

Report Structure:

- Section 4.1 — Basic detection across varying seasonal strengths
- Section 4.2 — Robustness to non-linear trends
- Section 4.3 — Performance across different trend types
- Section 5 — Additional real-world challenges (red noise, multiple seasonalities, amplitude modulation, outliers)

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:

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

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

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

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 $y_1, y_2, \ldots, y_n$, 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.

4 Simulation Studies

4.1 Simulation 1: Varying Seasonal Strength

4.1.1 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 = no seasonality, 1 = full seasonality).

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

4.1.2 Code

```r
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)
}
```

4.1.3 Example Time Series

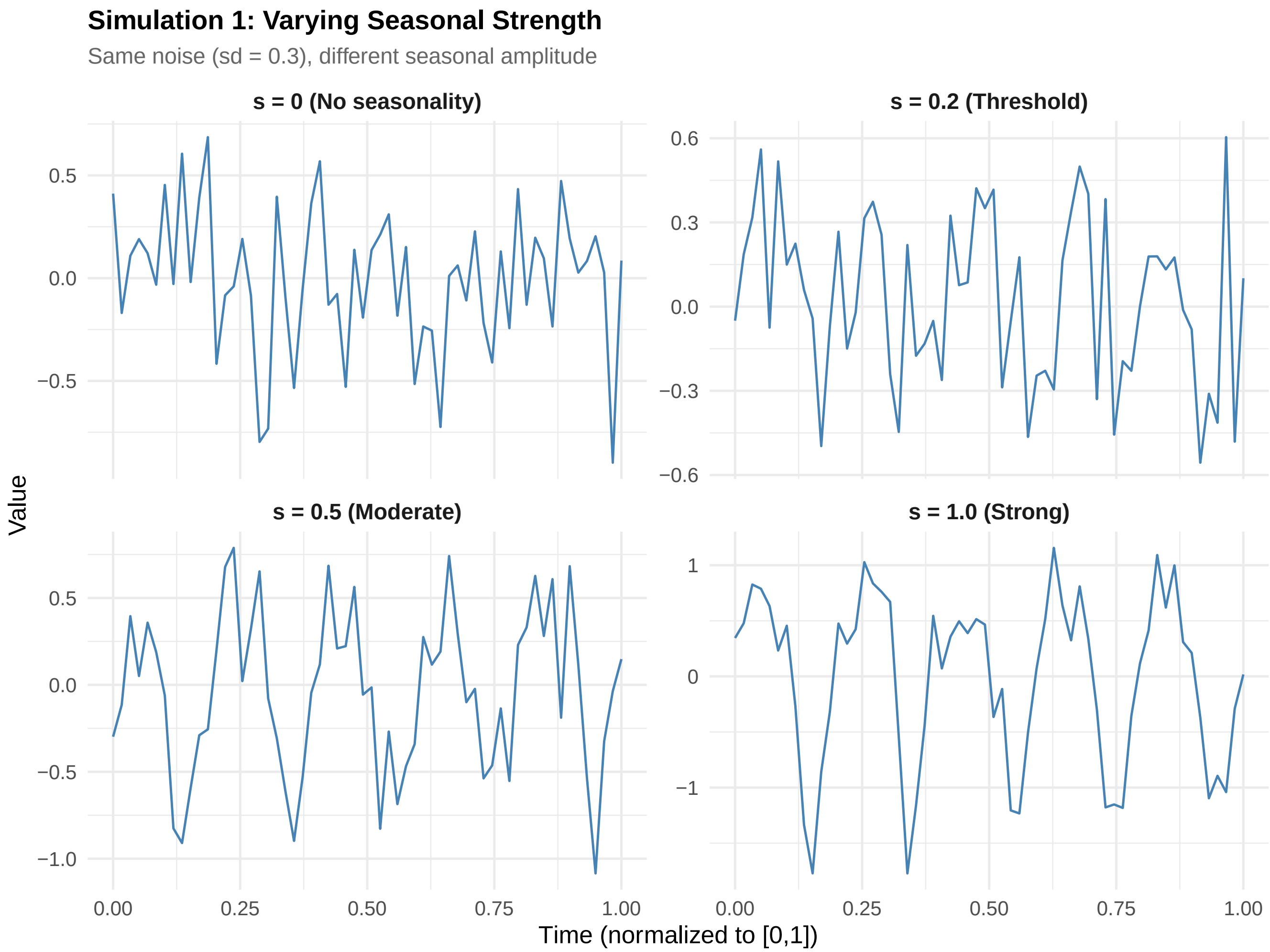

Figure 1: Example curves at different seasonal strength levels

4.1.4 Results

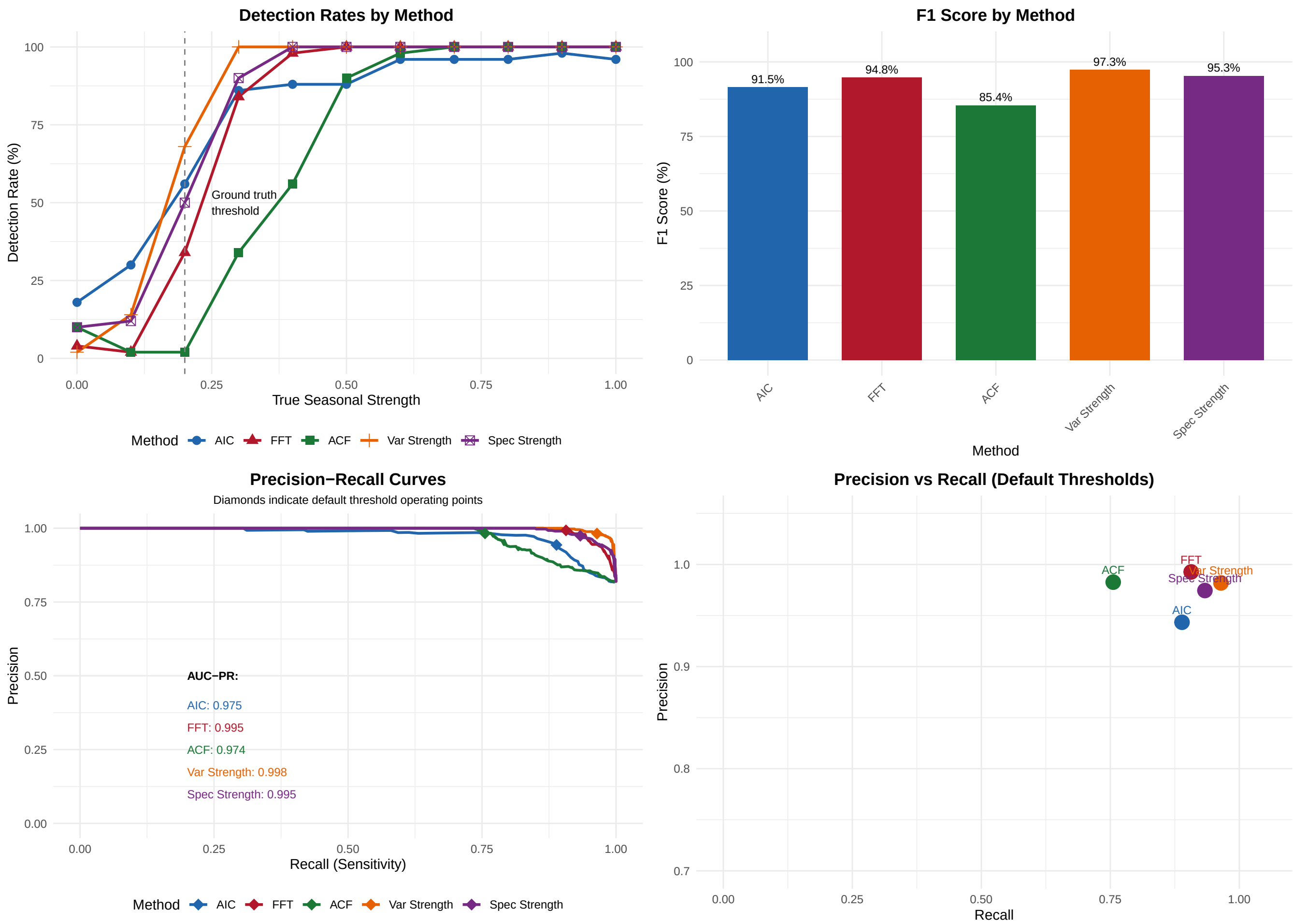

Figure 2: Detection rates by seasonal strength

How to interpret:

- The x-axis shows the true seasonal strength (0 = pure noise, 1 = 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

4.1.5 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?

4.1.6 Precision-Recall Analysis

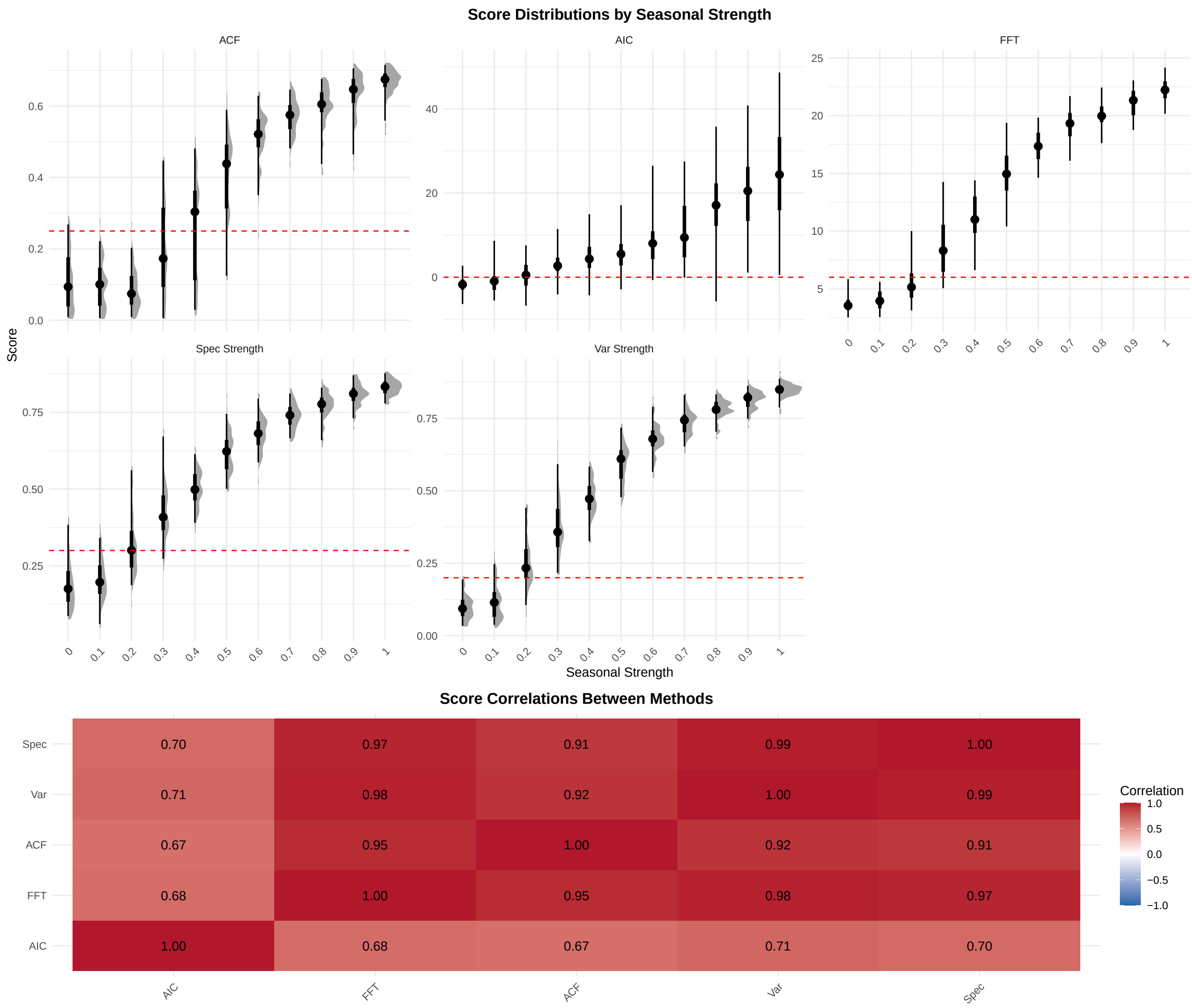

Figure 3: Precision-Recall curves

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

4.2 Simulation 2: Non-linear Trend

4.2.1 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:

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

4.2.2 Code

```r
# 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)
}
```

4.2.3 Example Time Series

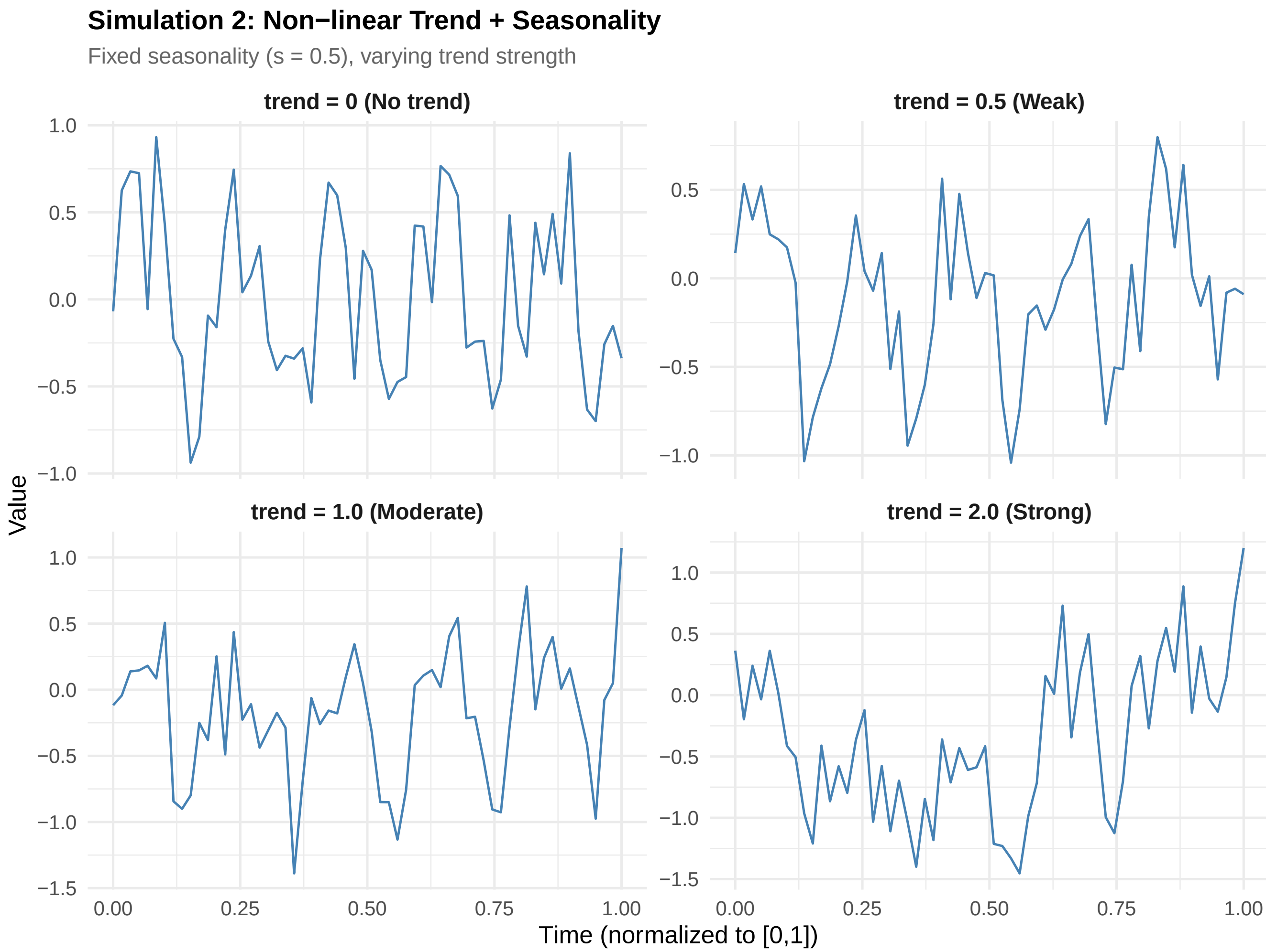

Figure 4: Example curves with fixed seasonality (s=0.5) and varying trend strength

4.2.4 Results

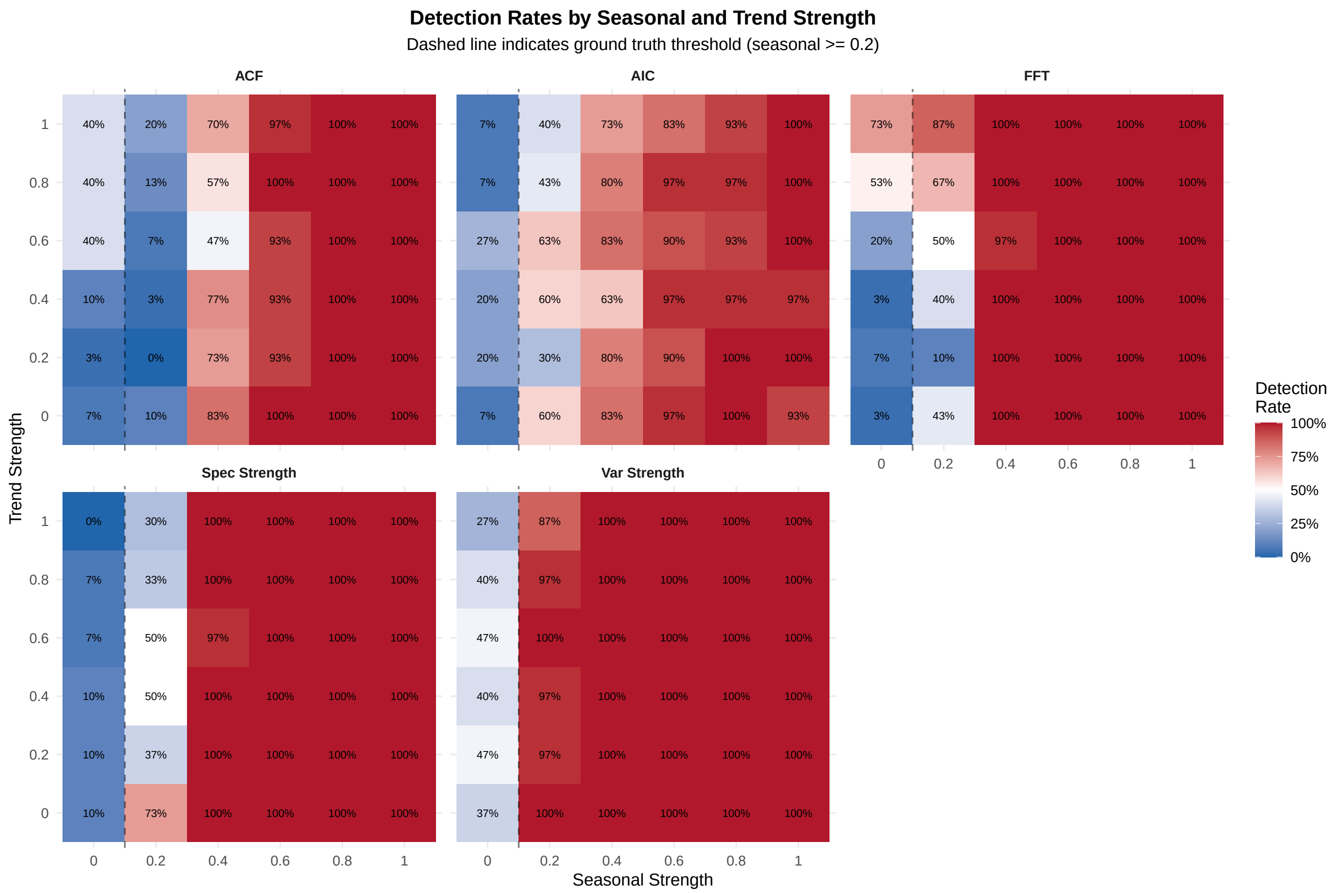

Figure 5: Detection rates heatmap by seasonal and trend strength

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

4.2.5 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

4.2.6 False Positive Rate by Trend Strength

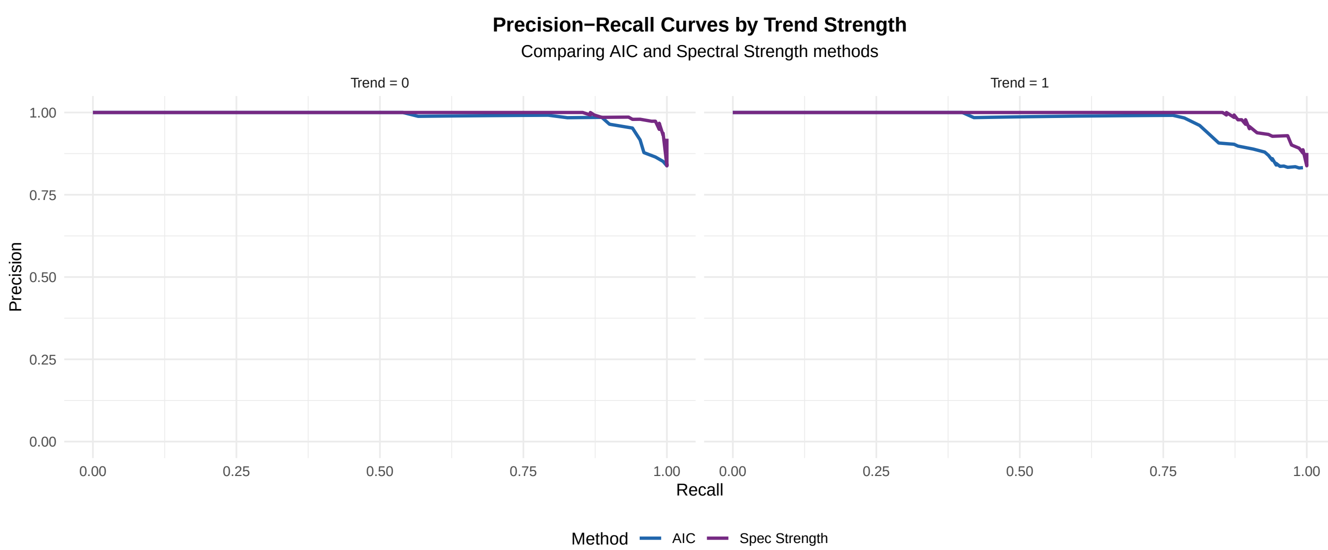

Figure 6: FPR when no seasonality is present, across trend strengths

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

4.3 Simulation 3: Multiple Trend Types

4.3.1 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

4.3.2 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)
)
```

4.3.3 Example Time Series

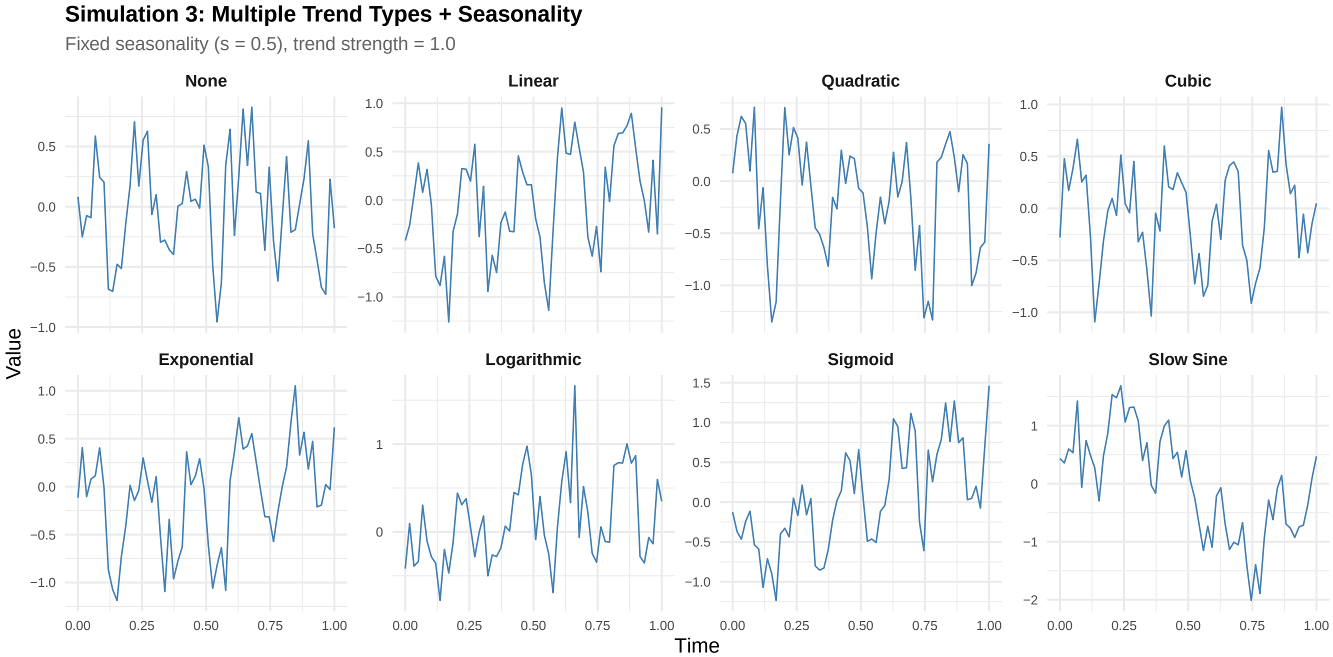

Figure 7: Example curves combining each trend type with seasonality (s=0.5, trend=1.0)

4.3.4 Results

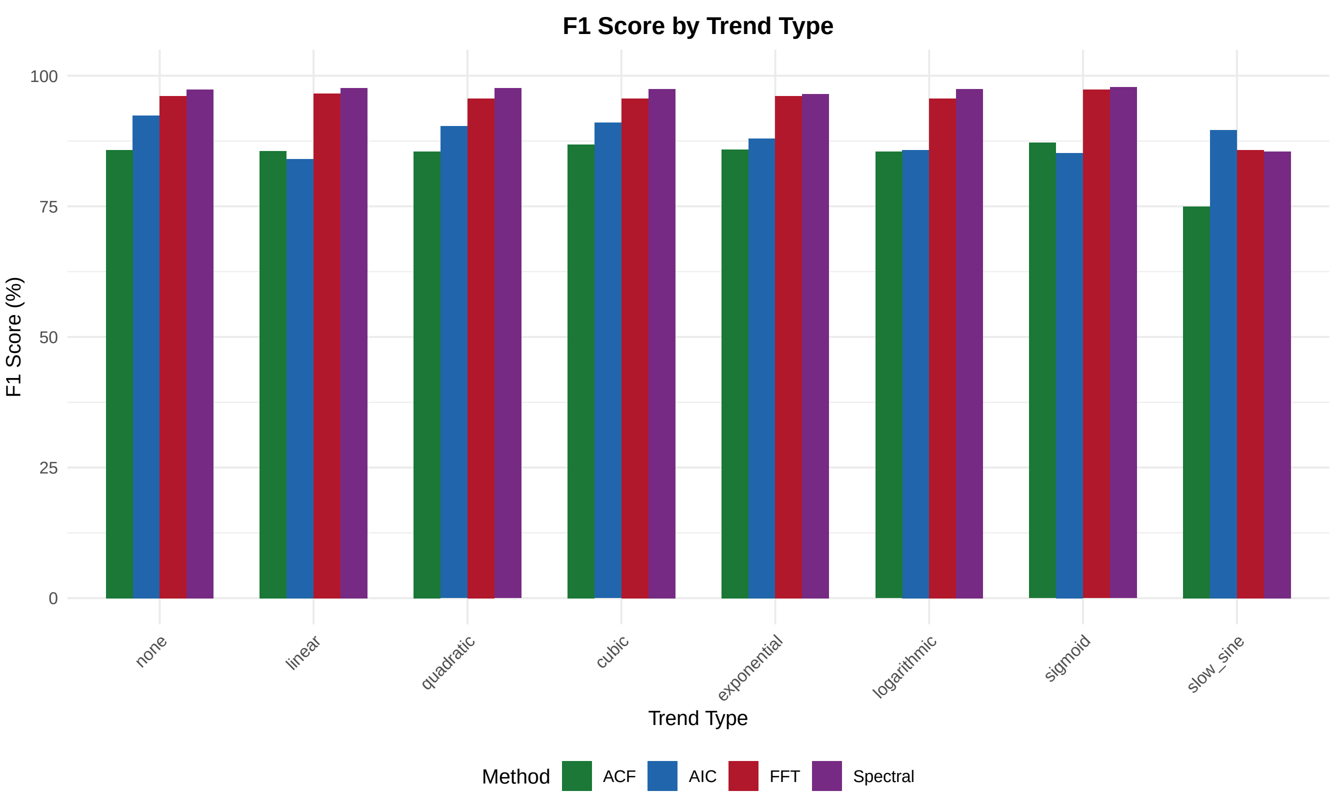

Figure 8: F1 scores by trend type

4.3.5 FPR by Trend Type

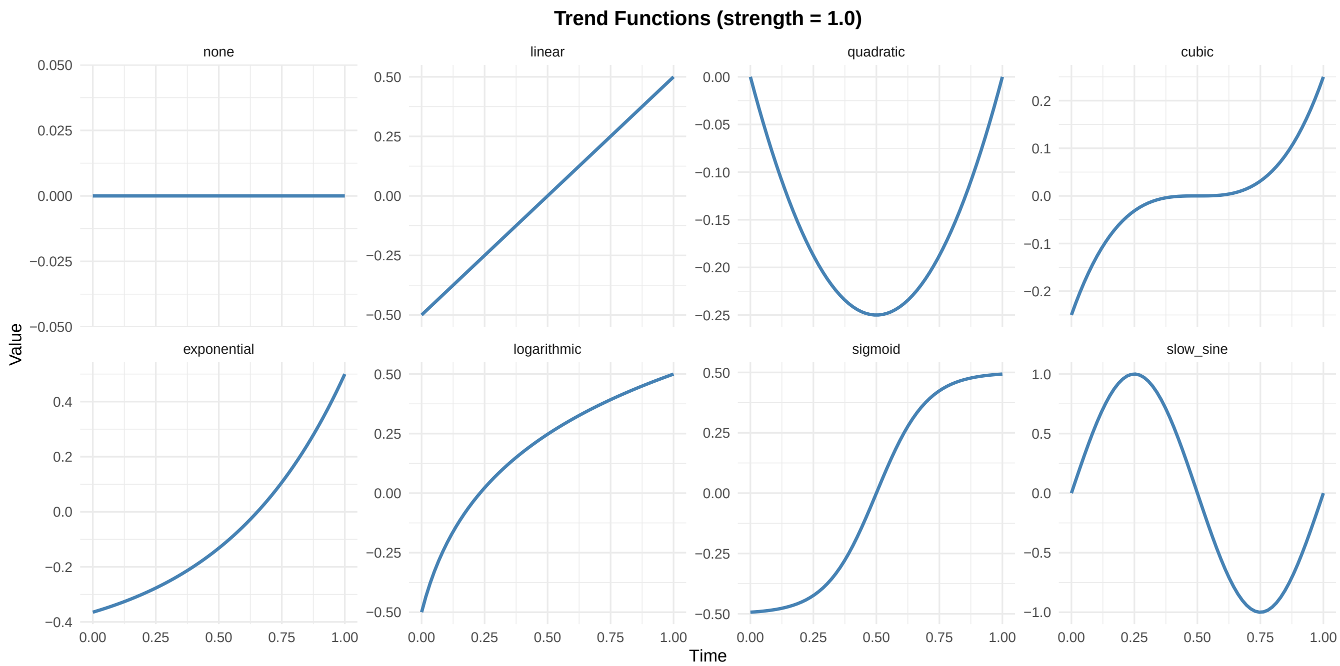

Figure 9: Example trend functions

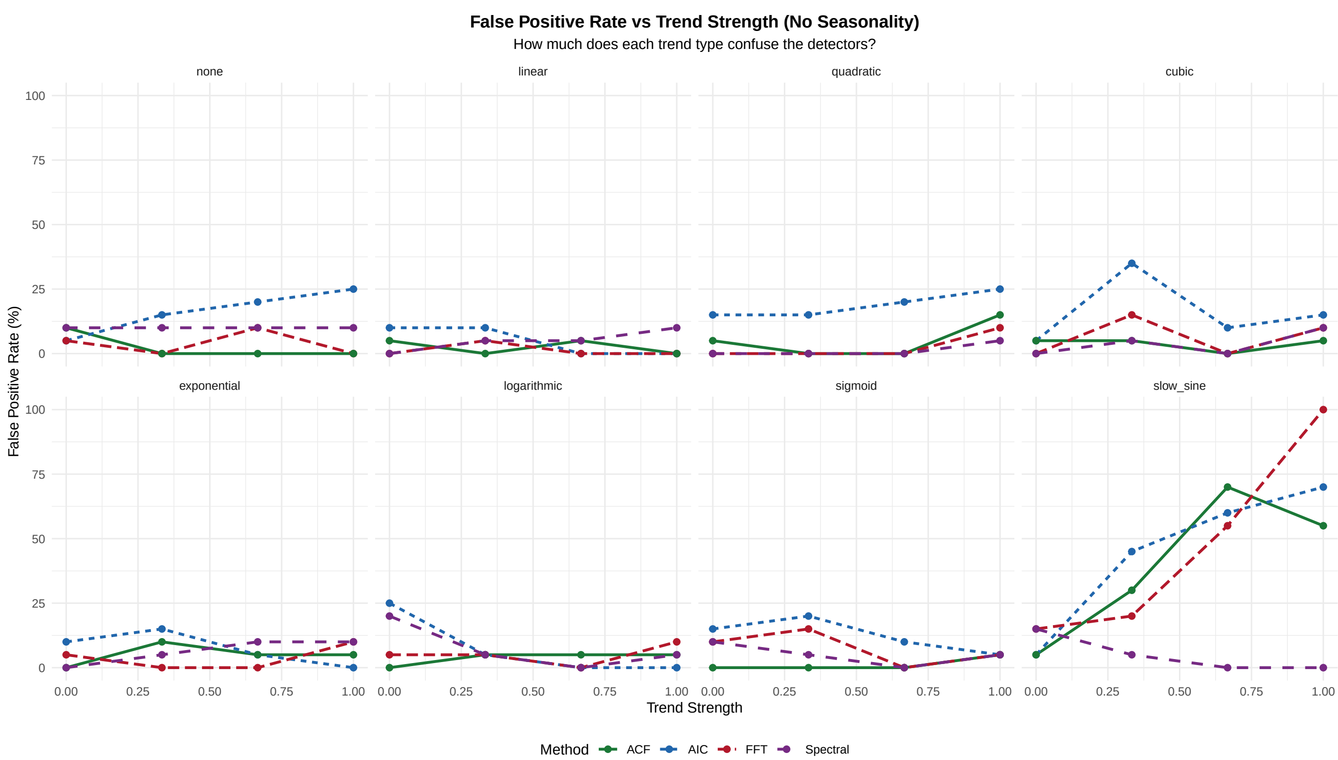

Figure 10: FPR across trend types and strengths

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 "seasonality"

4.3.6 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

5 Additional Robustness Challenges

The current simulations cover varying seasonal strength, non-linear trends, and multiple trend types. However, real-world data often breaks the assumptions made in these idealized setups. This section outlines additional robustness tests that would further validate method performance.

5.1 A. Colored Noise (Red Noise)

5.1.1 The Gap

The simulations currently use **white noise**: $\epsilon \sim N(0, 0.3^2)$ — independent, identically distributed.

5.1.2 Reality

Most physical and economic time series exhibit **red noise** (autocorrelated noise). Temperature records, stock prices, and sensor readings typically have positive autocorrelation where consecutive observations are more similar than distant ones.

AR(1) noise model:

$$\epsilon_t = \phi \epsilon_{t-1} + \eta_t, \quad \eta_t \sim N(0, \sigma^2)$$

where $\phi \in (0, 1)$ controls the autocorrelation strength.

5.1.3 Why It Matters

Spectral methods (FFT, Spectral Strength) and ACF are notoriously prone to **false positives** in red noise environments:

- A slow random walk can appear as a “trend” or low-frequency cycle
- AR(1) processes have elevated low-frequency power, mimicking seasonality
- The periodogram of red noise is not flat — it decays as $\sim 1/f^2$

Recommended test: Generate AR(1) noise with $\phi \in \{0.3, 0.5, 0.7, 0.9\}$ and measure FPR for each detection method.

5.2 B. Multiple Seasonalities

5.2.1 The Gap

The simulations assume a **single fixed period** (5 cycles over the observation window).

5.2.2 Reality

Data often contains **multiple nested seasonalities**:

- Energy consumption: daily cycle + weekly cycle + annual cycle
- Retail sales: weekly patterns + monthly patterns + holiday effects
- Traffic data: hourly patterns + daily patterns

Multi-seasonal model:

$$y(t) = s_1 \sin(2\pi f_1 t) + s_2 \sin(2\pi f_2 t) + \epsilon$$

where f_1 and f_2 are different frequencies (e.g., $f_1 = 5$ cycles, $f_2 = 20$ cycles).

5.2.3 Why It Matters

- Can **FFT** and **Spectral Strength** distinguish between a dominant seasonal frequency and secondary harmonics?
- Does **Variance Strength** get diluted when multiple periods are present but only one is specified?
- What happens when the specified period misses the dominant frequency?

Recommended test: Generate signals with primary period p_1 and secondary period p_2 with varying amplitude ratios, then test detection with period set to p_1.

5.3 C. Amplitude Modulation (Time-Varying Seasonality)

5.3.1 The Gap

The signal model assumes **constant seasonal strength** s across the entire curve:

$$y(t) = s \cdot \sin(2\pi f t) + \epsilon$$

5.3.2 Reality

Seasonality often **grows or shrinks over time** (multiplicative seasonality):

- Heating demand: seasonal amplitude is higher in extreme years
- Economic growth: seasonal patterns amplify as the economy scales
- Agricultural yields: seasonal variation depends on climate conditions that vary year-to-year

Amplitude-modulated model:

$$y(t) = s(t) \cdot \sin(2\pi f t) + \epsilon$$

where $s(t)$ is a time-varying envelope, e.g., $s(t) = s_0 \cdot (1 + \alpha t)$ (linear growth) or $s(t) = s_0 \cdot (1 + \beta \sin(2\pi t/T))$ (periodic modulation).

5.3.3 Why It Matters

- **Variance Strength** averages variance globally — it may under-report seasonality if the signal is strong in only half the time domain
- Methods might fail when seasonality "emerges" partway through the series
- Detection thresholds calibrated on constant-amplitude signals may be inappropriate

Recommended test: Generate signals where seasonal amplitude varies from 0→1 across the series, and test whether methods detect "partial seasonality."

5.4 D. Outliers and Anomalies

5.4.1 The Gap

The current noise model is **Gaussian** with constant variance.

5.4.2 Reality

Real sensors and measurements have:

- **Spikes**: Sudden large values (e.g., sensor glitches, recording errors)
- **Dropouts**: Missing or zero values
- **Level shifts**: Sudden changes in baseline (e.g., sensor recalibration)
- **Heavy tails**: Non-Gaussian error distributions

Contaminated noise model:

$$\epsilon_t = \begin{cases} N(0, \sigma^2) & \text{with probability } 1 - p \\ N(0, k^2\sigma^2) & \text{with probability } p \end{cases}$$

where p is the outlier probability and $k > 1$ is the outlier magnitude multiplier.

5.4.3 Why It Matters

- A **single large outlier** can distort the FFT spectrum, creating spurious peaks
- Outliers inflate $\text{Var}(R_t)$ in the Variance Strength denominator, potentially causing **false negatives**
- ACF is sensitive to outliers, which can destroy or create spurious autocorrelation

Recommended test: Add $p = 5\%$ outliers with magnitude $k = 5$ and measure degradation in F1 scores.

5.5 Summary of Robustness Gaps

Challenge	Current Assumption	Real-World Behavior	Methods at Risk
Red Noise	White noise (i.i.d.)	Autocorrelated (AR)	FFT, ACF, Spectral
Multiple Seasonalities	Single period	Nested periods	All (period misspecification)
Amplitude Modulation	Constant strength	Time-varying strength	Variance (averaging)
Outliers	Gaussian	Heavy-tailed, spikes	FFT, Variance, ACF

5.6 Experimental Results

We ran simulations for each robustness scenario. Key findings:

5.6.1 A. Red Noise Results

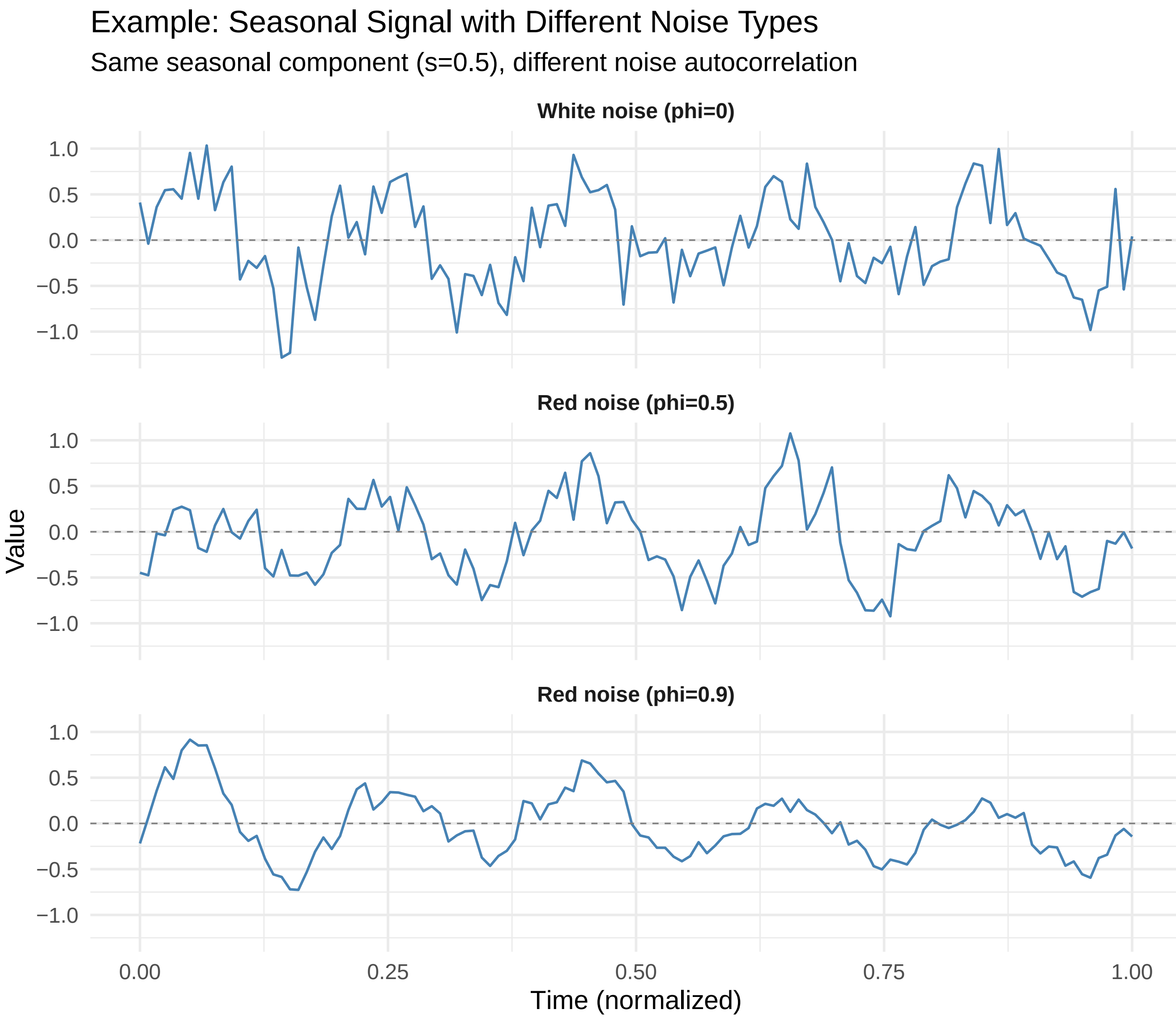

Figure 11: Example time series with different noise types

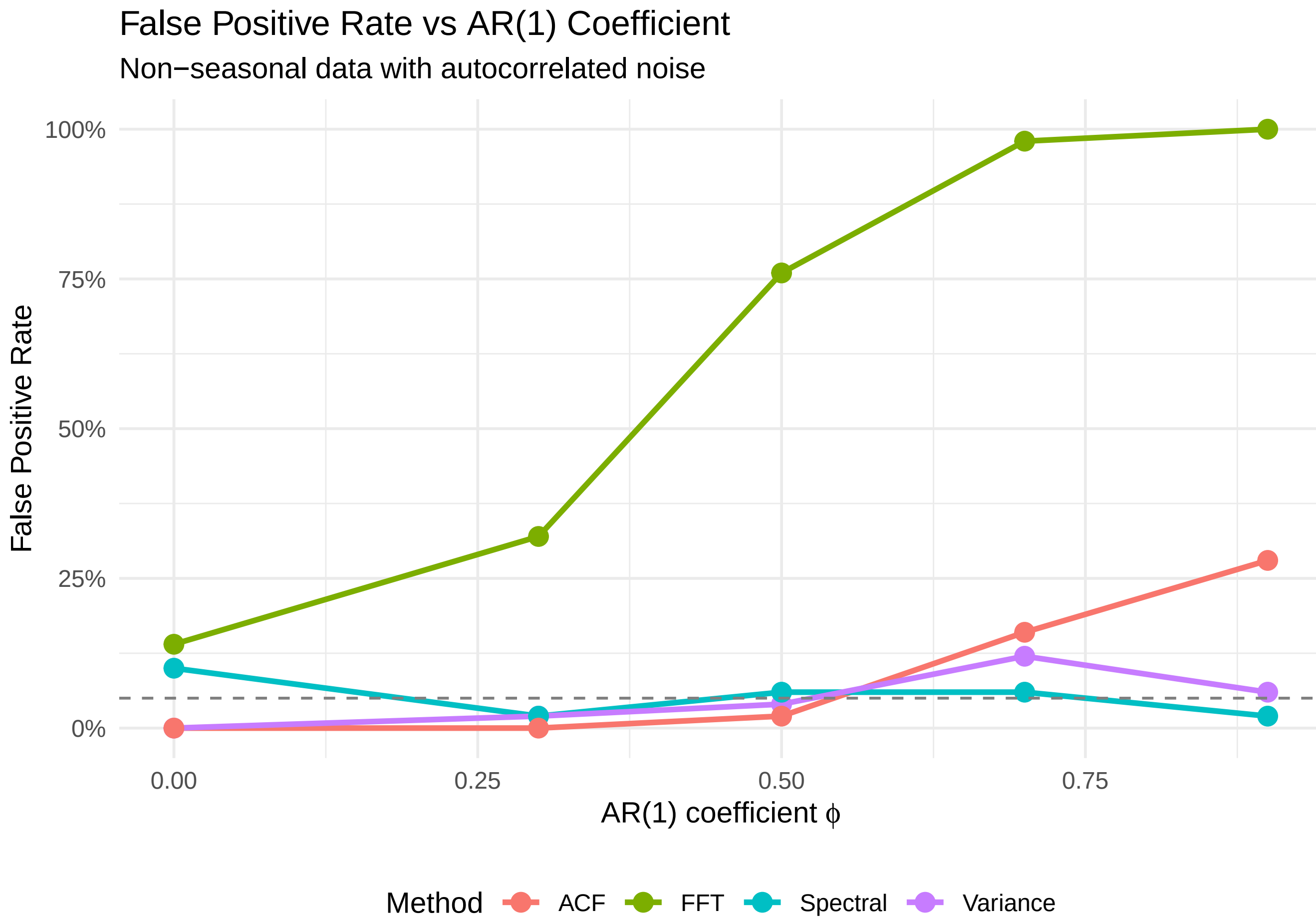

Figure 12: False positive rate vs AR(1) coefficient

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

Key finding: FFT is catastrophically affected by red noise, with FPR reaching 100% at high autocorrelation. Variance and Spectral methods remain relatively robust.

5.6.2 B. Multiple Seasonalities Results

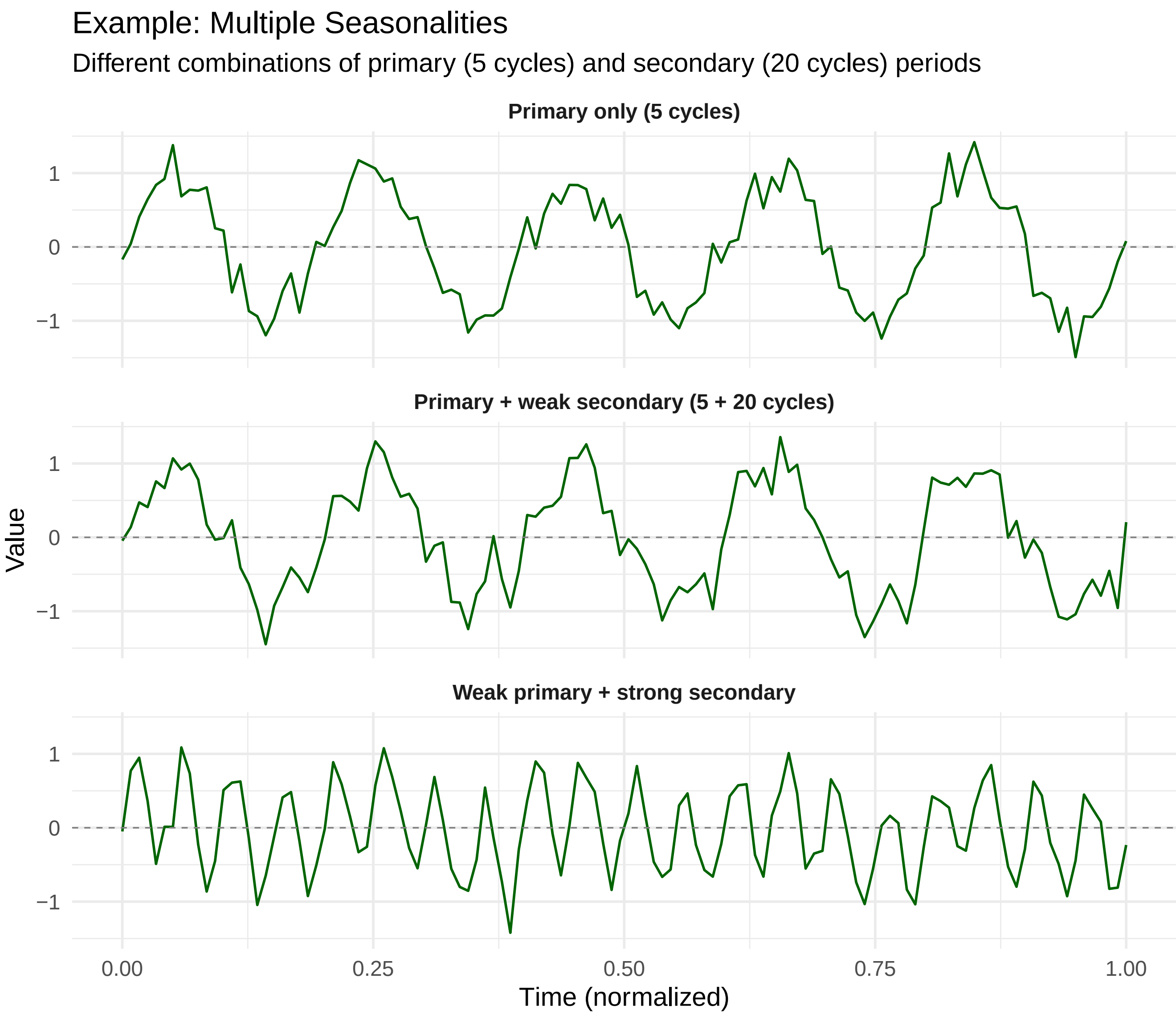

Figure 13: Example time series with multiple seasonal components

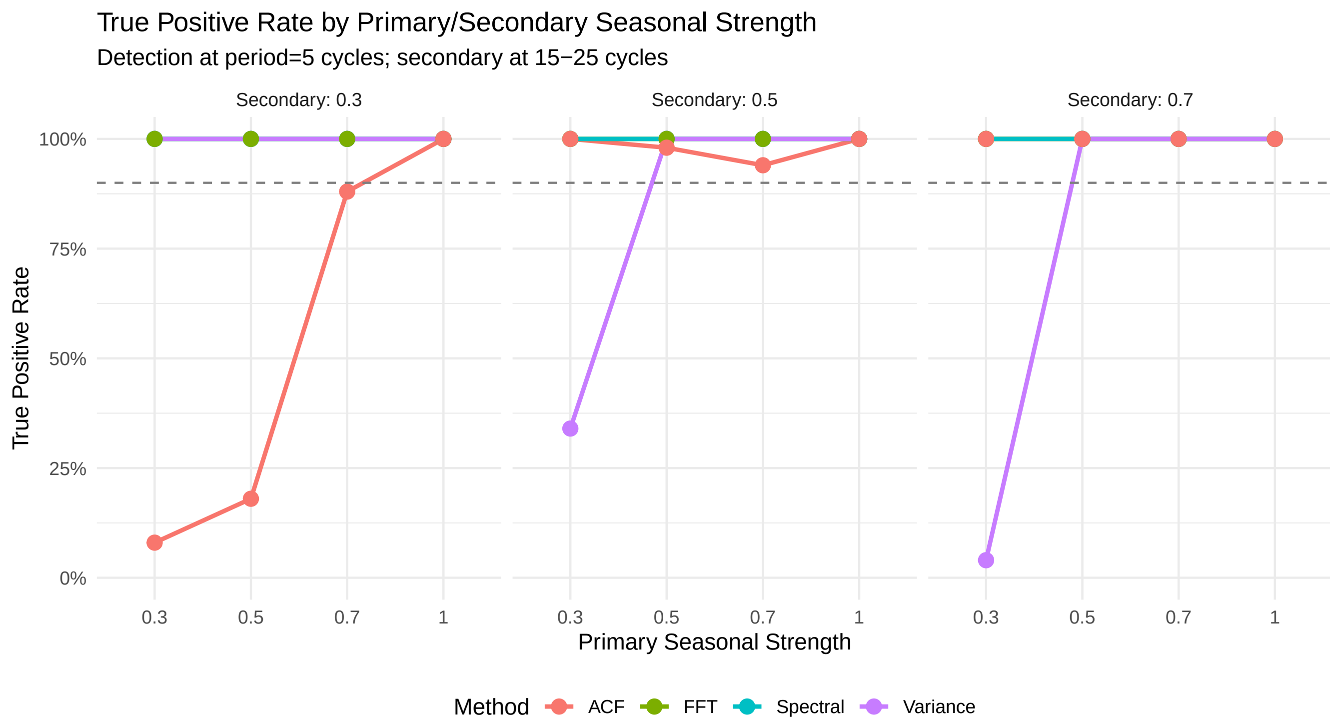

Figure 14: TPR by primary and secondary seasonal strength

When secondary seasonality dominates (strength 0.7 vs primary 0.3):

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

Key finding: Variance Strength fails when secondary seasonality dominates because period is misspecified; FFT/Spectral detect any periodicity regardless of which component dominates.

5.6.3 C. Amplitude Modulation Results

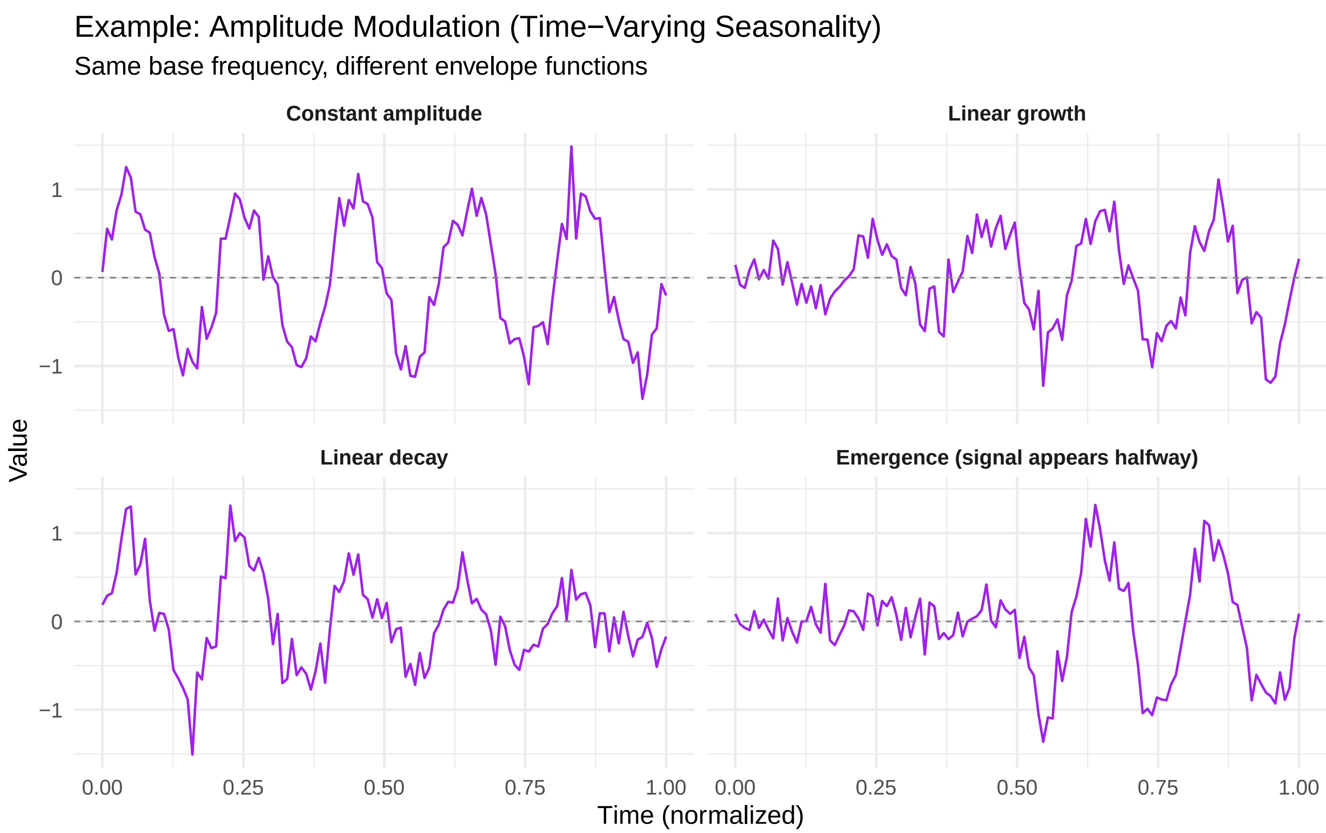

Figure 15: Example time series with different amplitude modulation types

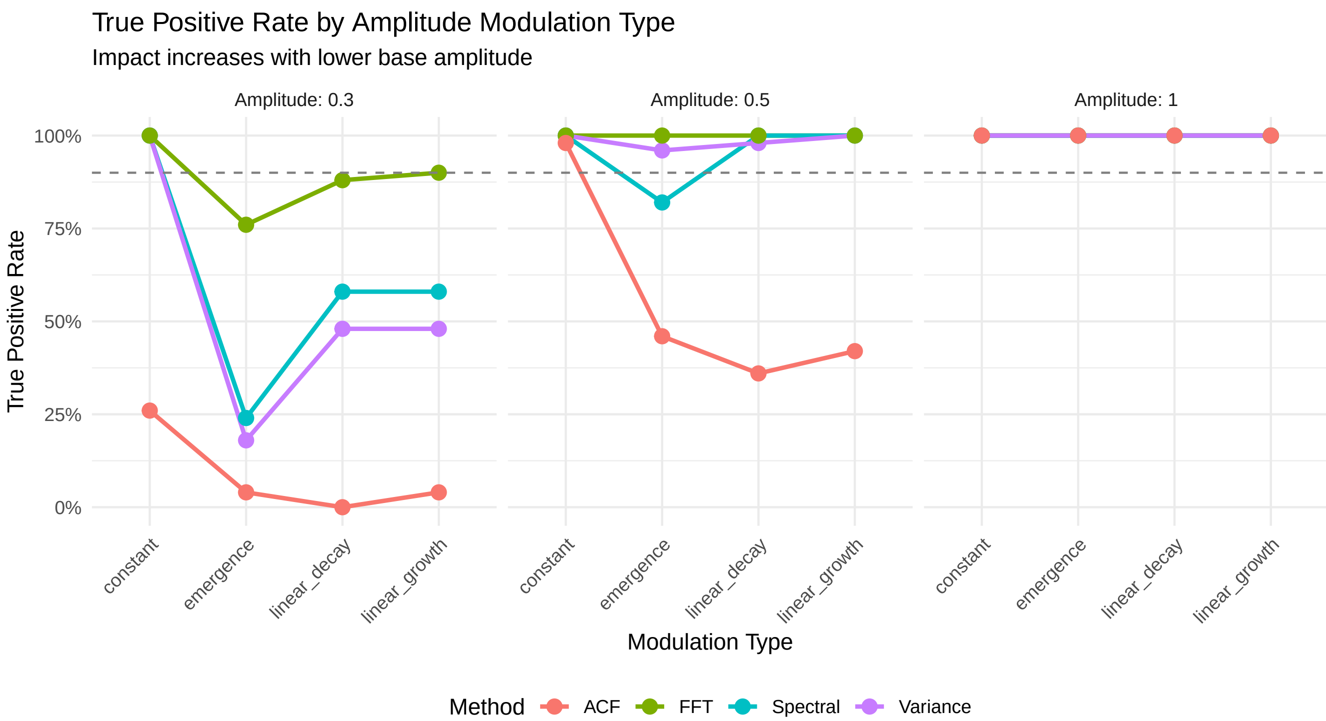

Figure 16: TPR by amplitude modulation type and base amplitude

At base amplitude 0.3:

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

Key finding: "Emergence" pattern (signal only in second half) is most challenging. FFT remains most robust to modulation.

5.6.4 D. Outlier Results

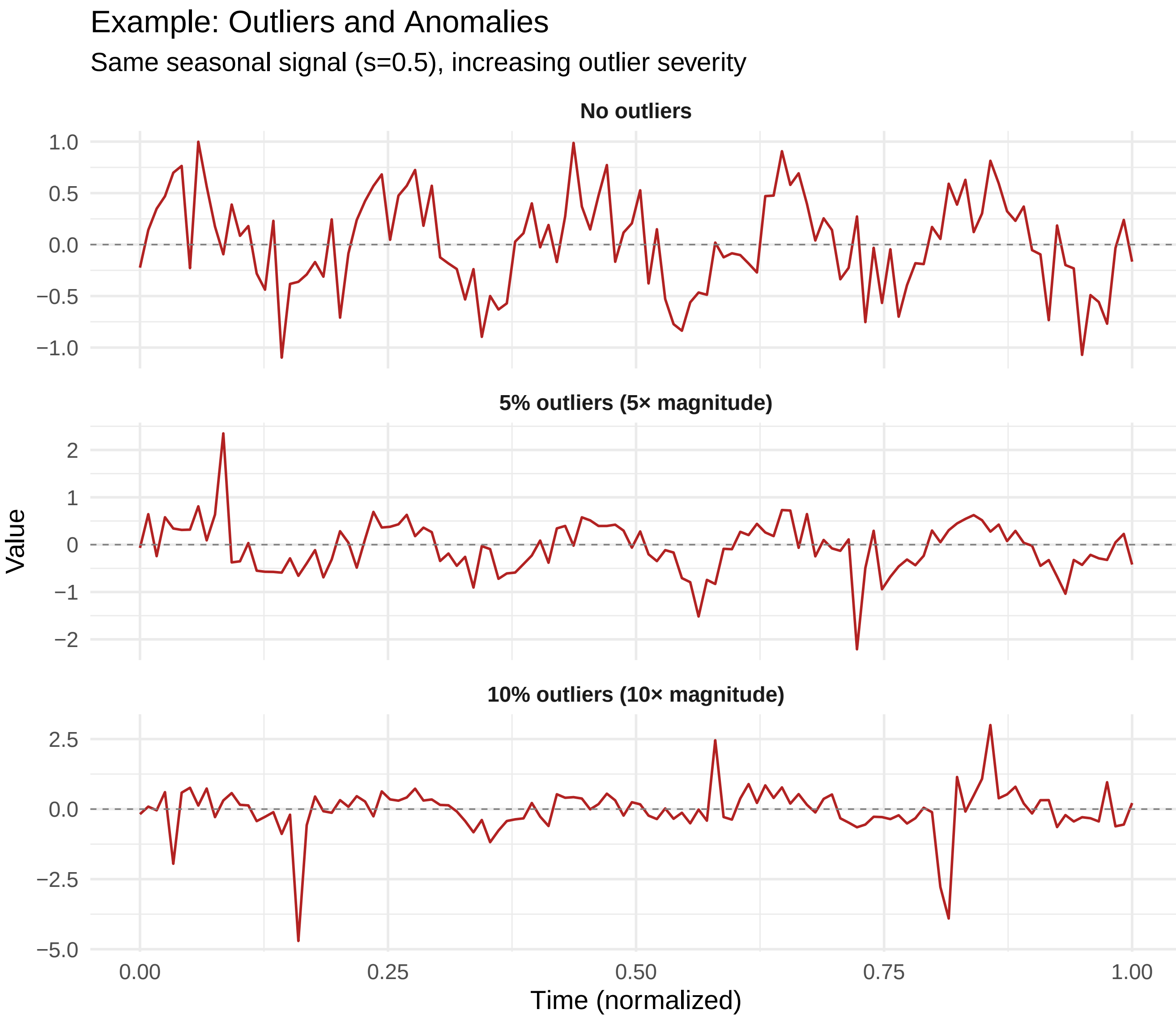

Figure 17: Example time series with different outlier severities

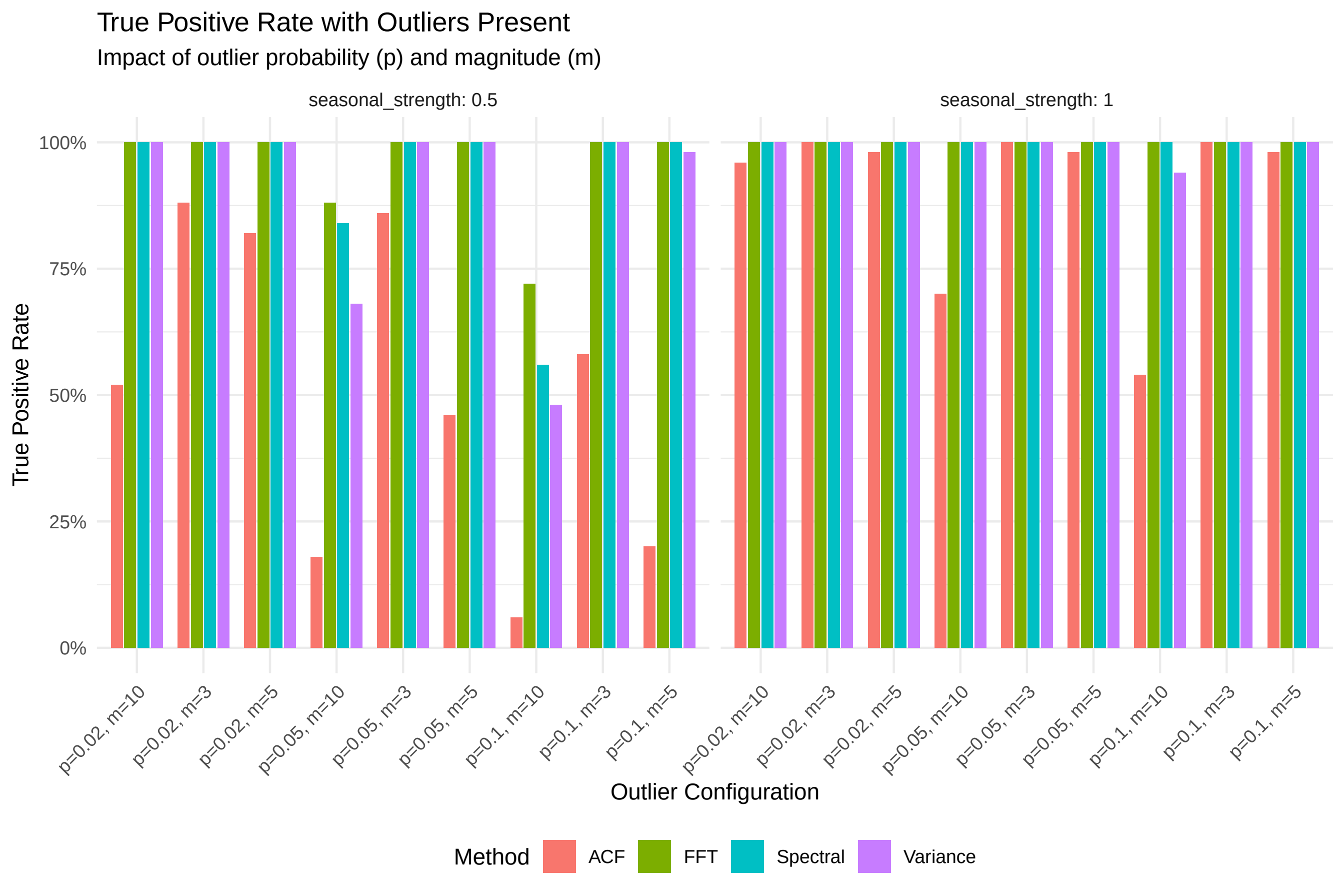

Figure 18: Impact of outliers on TPR

TPR for seasonal data (s=0.5) with increasing outlier severity:

Outliers	Magnitude	Variance	Spectral	FFT	ACF
2%	5×	100%	100%	100%	70%
5%	5×	100%	98%	100%	48%
5%	10×	66%	78%	88%	14%
10%	5×	92%	94%	98%	20%
10%	10×	**40%**	52%	66%	**2%**

Key finding: ACF is most sensitive to outliers; Variance Strength degrades significantly at high outlier severity (10%, 10×). Pre-filtering outliers recommended for real data.

6 Key Findings

6.1 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

6.2 Critical Issues Found

1. **Period units matter**: The `period` parameter in `seasonal_strength()` must be in argvals units, not raw time units (e.g., 0.2 not 12)
2. **FFT is vulnerable to slow oscillations**: Any periodic signal (even non-seasonal) triggers detection

7 Recommendations

7.1 For Unknown Datasets

Primary recommendation: Variance Strength

```r
# 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
strength <- seasonal_strength(fd,
                              period = period_in_argvals_units,
                              method = "spectral",
                              detrend = "linear")
is_seasonal <- strength > 0.3
```

Ensemble approach (most robust):

```r
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
```

7.2 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 $\rightarrow$ positive difference

Calibration methodology: All thresholds were calibrated using pure noise data (seasonal strength = 0, no trend) by taking the 95th percentile of each method's score distribution. This ensures approximately 5% false positive rate on clean data. Note that FPR may increase when confounding trends are present (see Simulation 2 and 3 results).

8 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.

8.1 Limitations and Future Work

The current simulations use idealized conditions (white noise, single seasonality, constant amplitude). Real-world data presents additional challenges:

- **Red noise** (autocorrelated errors) can inflate false positives for spectral methods
- **Multiple seasonalities** require careful period specification
- **Amplitude modulation** may dilute variance-based measures
- **Outliers** can distort all methods

See Section 5 for detailed discussion of these robustness challenges and recommended tests.

9 Appendix: Fourier vs P-spline Comparison

The AIC comparison method is based on the observation that Fourier bases naturally capture periodic patterns better than P-splines for seasonal data.

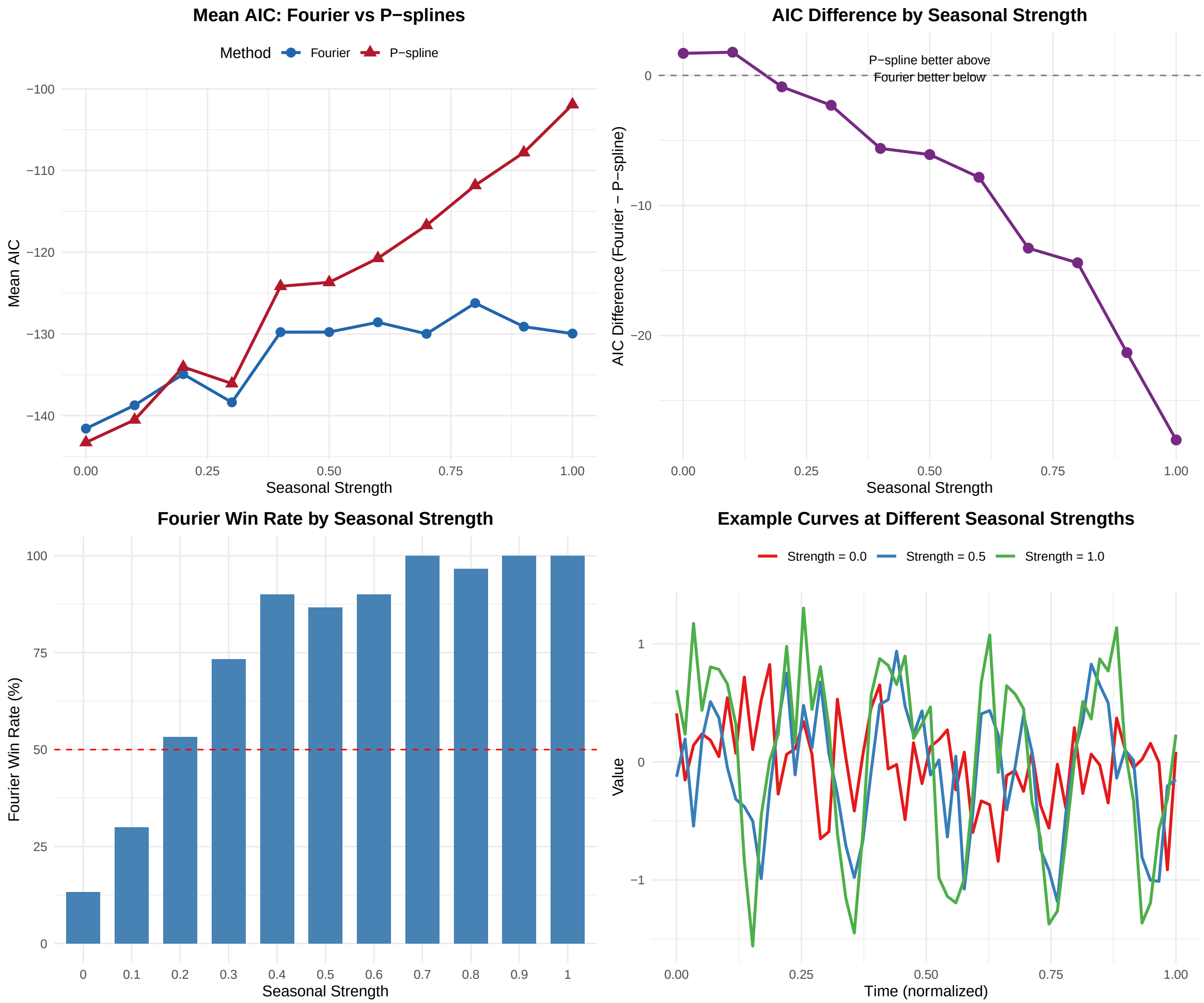

Figure 19: Fourier vs P-spline AIC comparison

How to interpret:

- Top left: Mean AIC for both methods across seasonal strengths
- Top right: AIC difference (Fourier - P-spline) showing crossover point
- Bottom left: Fourier win rate at each strength level
- Bottom right: Example curves at different seasonal strengths

10 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` — Fourier vs P-spline AIC study
- `seasonality_robustness_tests.R` — Robustness tests (red noise, multi-seasonal, etc.)
- `generate_training_data.R` — Generate training data for ML classifiers

PDF outputs are in the `plots/` subfolder.