July 20 Update

Simulating a Modulated Signal

From Kian's thesis we have the following formula for signals with frequency modulation:

$$X_t = \sum_m \mu_m cos(2\pi f_m t + 2\pi \int_0^t \phi_m(\tau) d\tau) + Z_t$$

Now let $\theta(t) = 2\pi f_m t + 2\pi \int_0^t \phi_m(\tau) d\tau$. Since ϕ and θ are both polynomials, we can simplify to:

$$\phi(\tau) = \sum_{p=0}^{P} a_p \tau^p$$

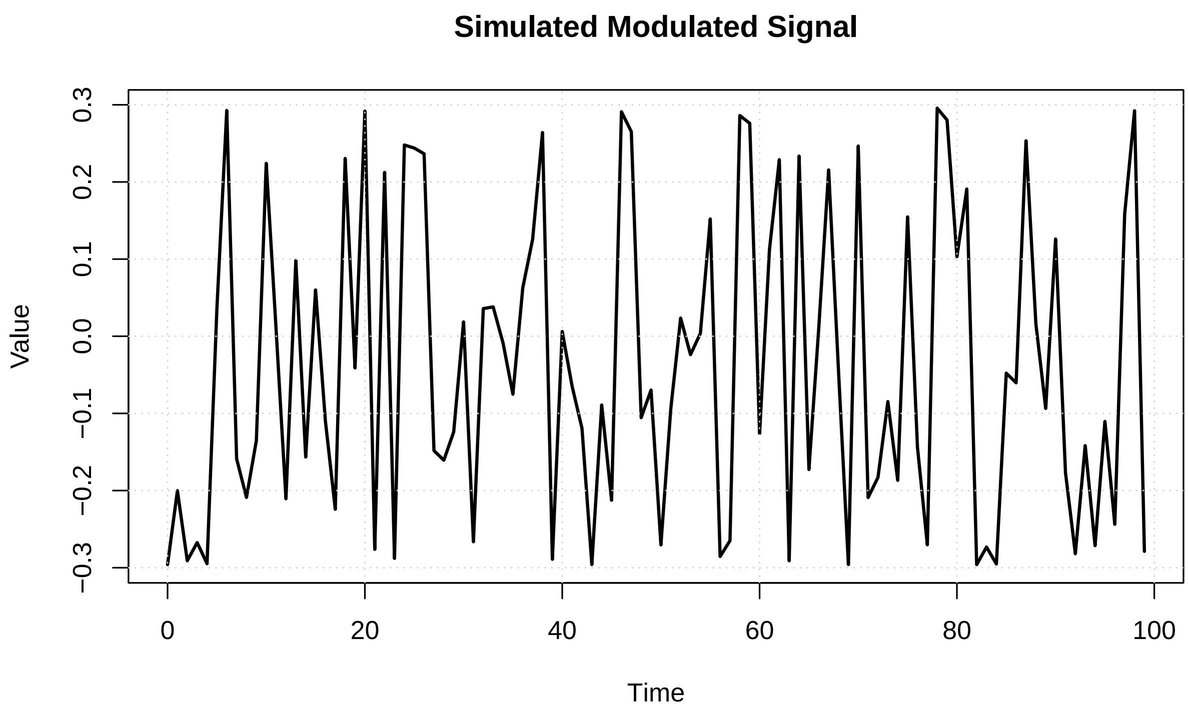
$$\theta(t) = 2\pi f_m t + 2\pi \int_0^t \sum_{p=0}^{P} a_p \tau^p d\tau$$

$$\theta(t) = 2\pi f_m t + 2\pi \sum_{p=0}^{P} \frac{a_p}{p+1} t^{p+1}$$

Using these formulas I wrote a function to simulate a modulated signal given a length N, number of frequencies m, and the maximim degree of polynomial to consider P.

```
## Simulating modulated signal
set.seed(11)
Tt = simTt_mod(N = 100, numFreq = 1, P = 2)

## Printing the corresponding summation
print(Tt$fn)
```

```
## [1] "(-0.296)*cos(2*pi*(1.635*t + (-1.098)*t^1 + (0.4235)*t^2 + (-0.0923333333333333)*t^3 + 0))"
```

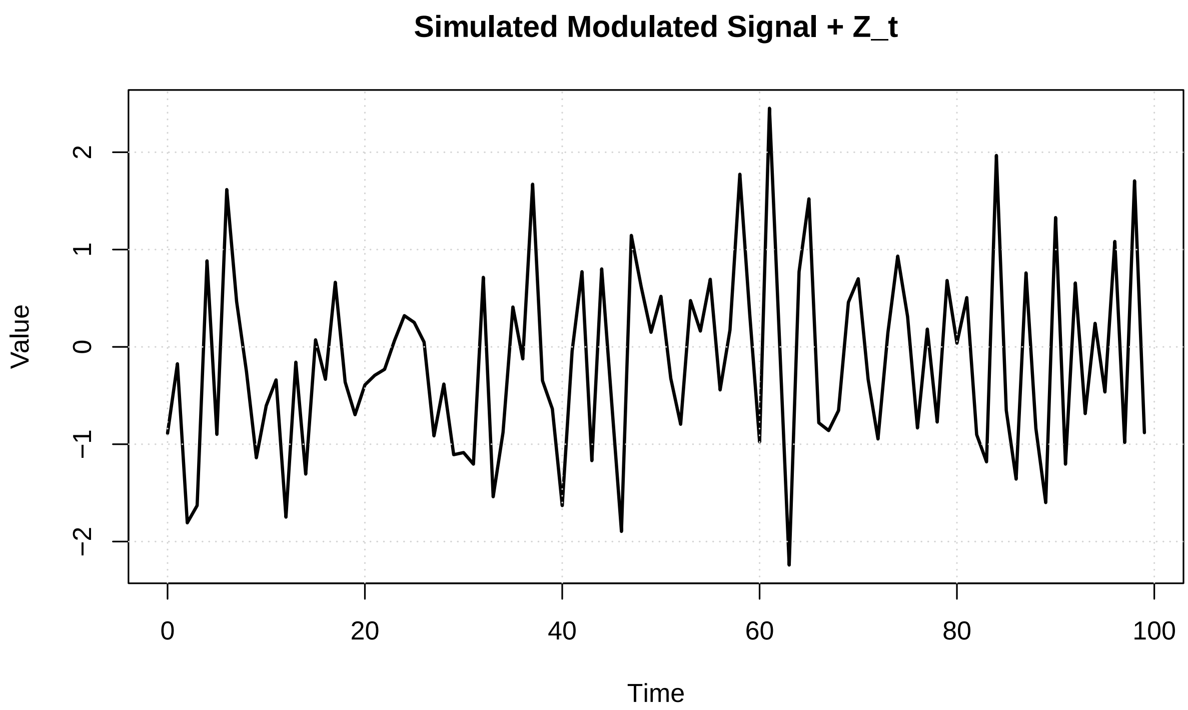


To generate values for μ, f, and a we use a similar idea to the interpTools "simTt" function:

```
## Defining "Fourier Frequency"
# fourierFreq = 2*pi/N

## Generating cosine coefficients
# mu = rnorm(1, mean = 0, sd = N/200)

## Generating frequencies
# f = runif(1, fourierFreq, pi)

## Generating "a" coefficients
# a = rnorm(P, mean = 0, sd = N/200)
```

Now using the interpTools "simWt" function, we can add some weakly-stationary noise to account for Z_t:

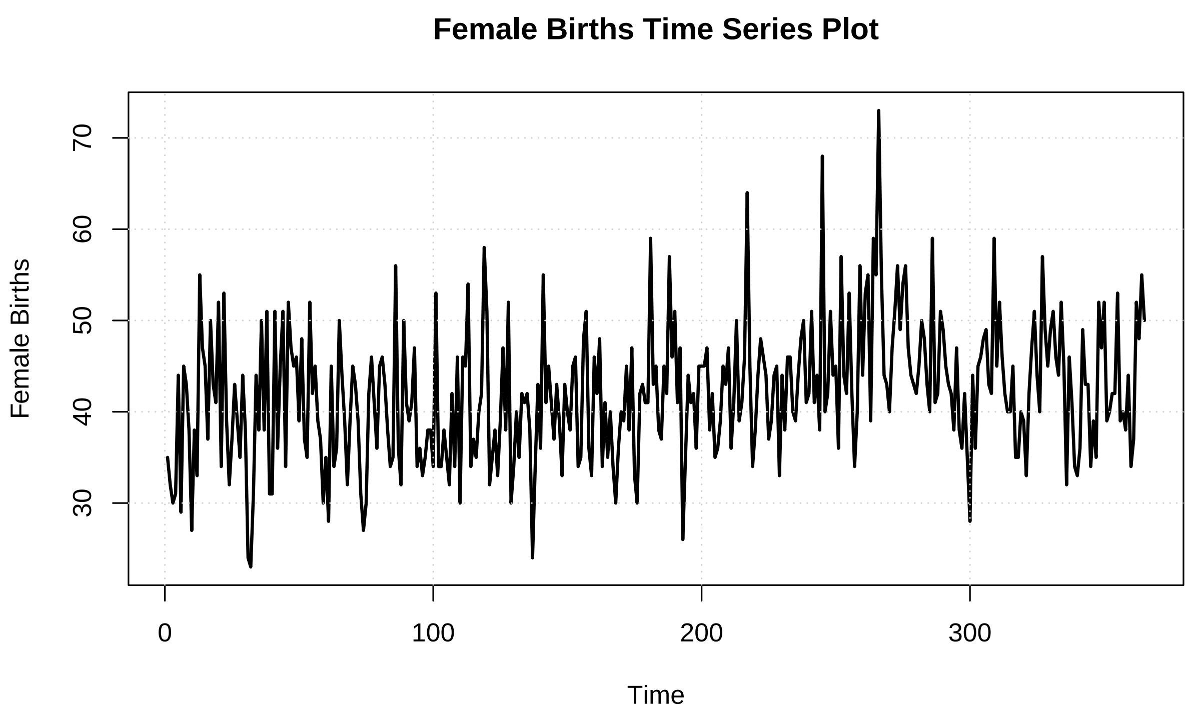
```
## Simulating noise from interpTools
set.seed(11)
Wt = interpTools::simWt(N = 100)$value
```


Simulated Modulated Signal + Z_t
Value
Time
2
1
0
−1
−2
0
20
40
60
80
100

Simulation Results

- Used three real-world time series
- Used one time series tailored for the Hybrid Weiner method
- Used one time series with a modulated signal
- Considered missing proportion (P) in [10%, 20%, 30%]
- Considered gap width (G) in [10, 25, 50]
- Each method is tested with 25 unique combinations of P and G
- Displayed results are averaged over the 25 iterations
- Considered log-cosh loss (LCL), MAE, and RMSE for evaluation
- Note: Neural nets are designed to minimize RMSE in training

Data Set 1:



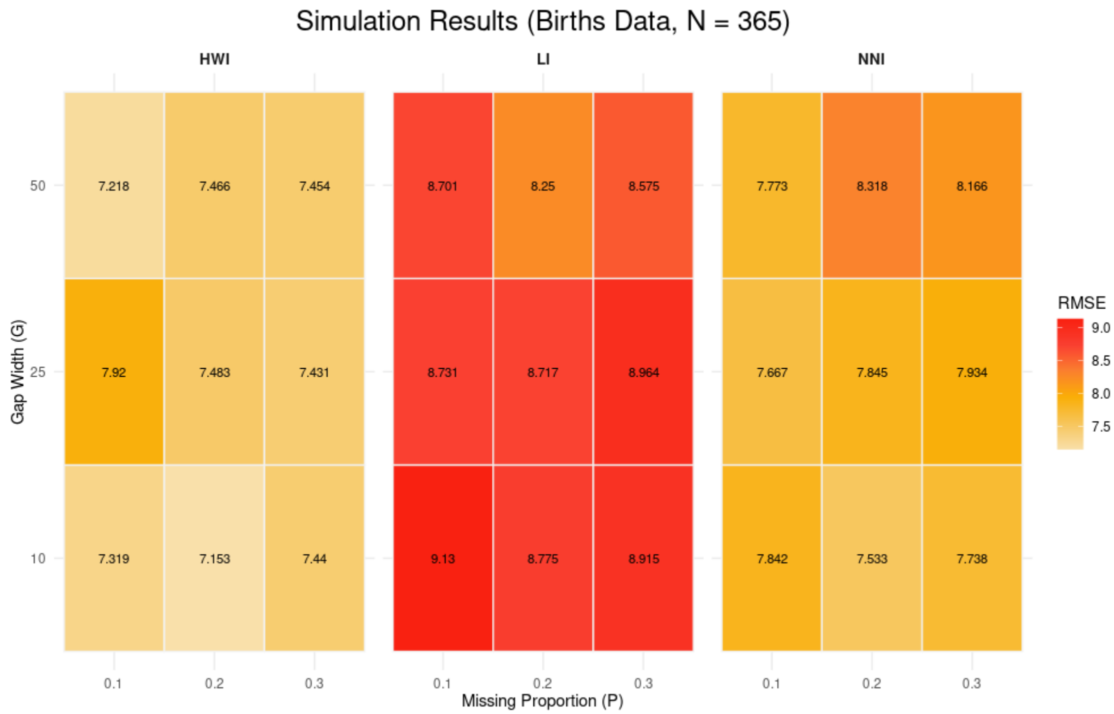
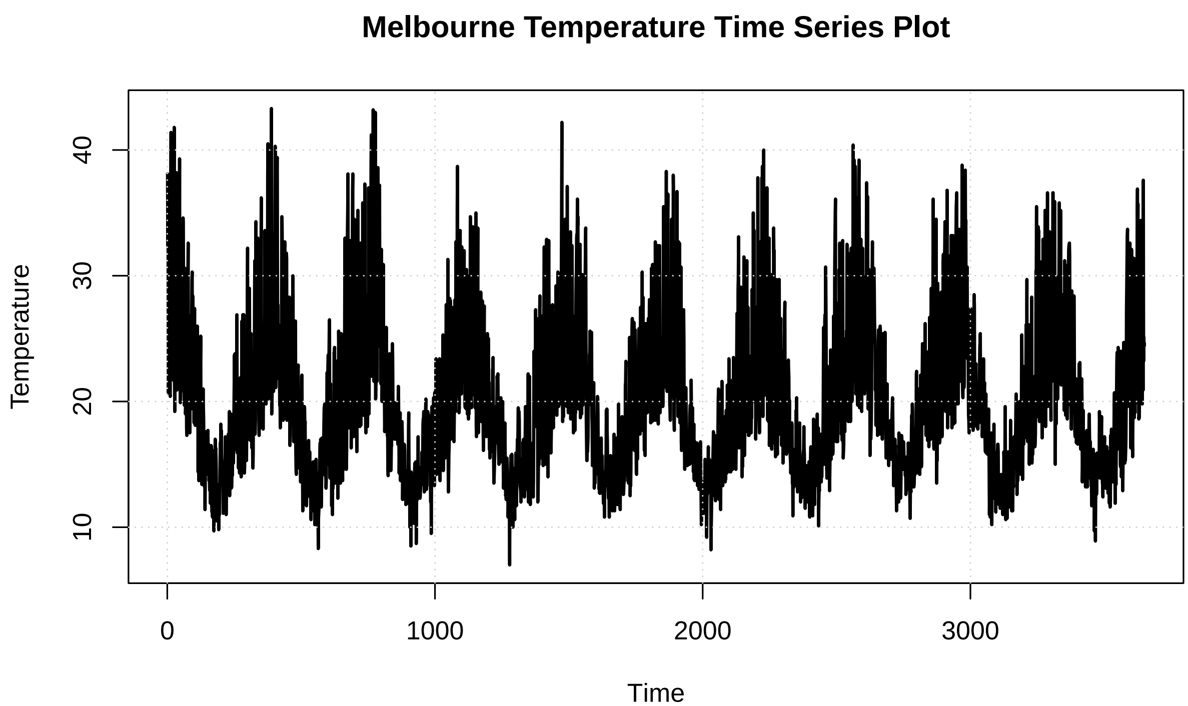


Figure 1: Births data RMSE performance

Data Set 2:



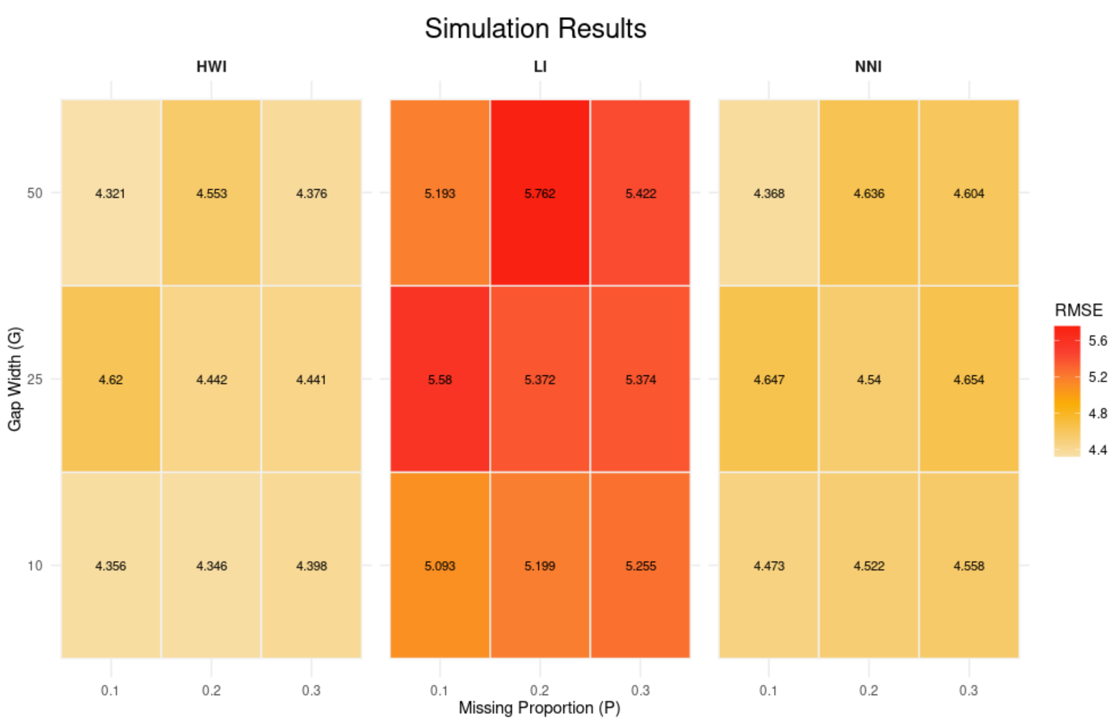
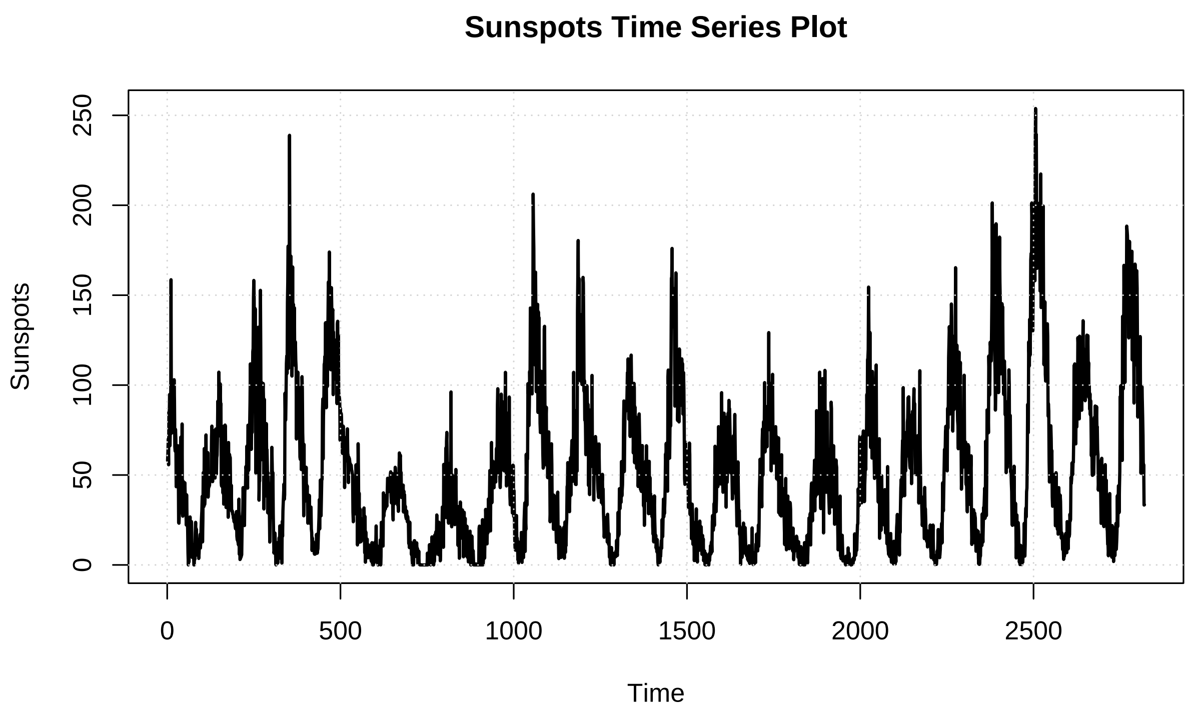


Figure 2: Temperature data RMSE performance

Data Set 3:



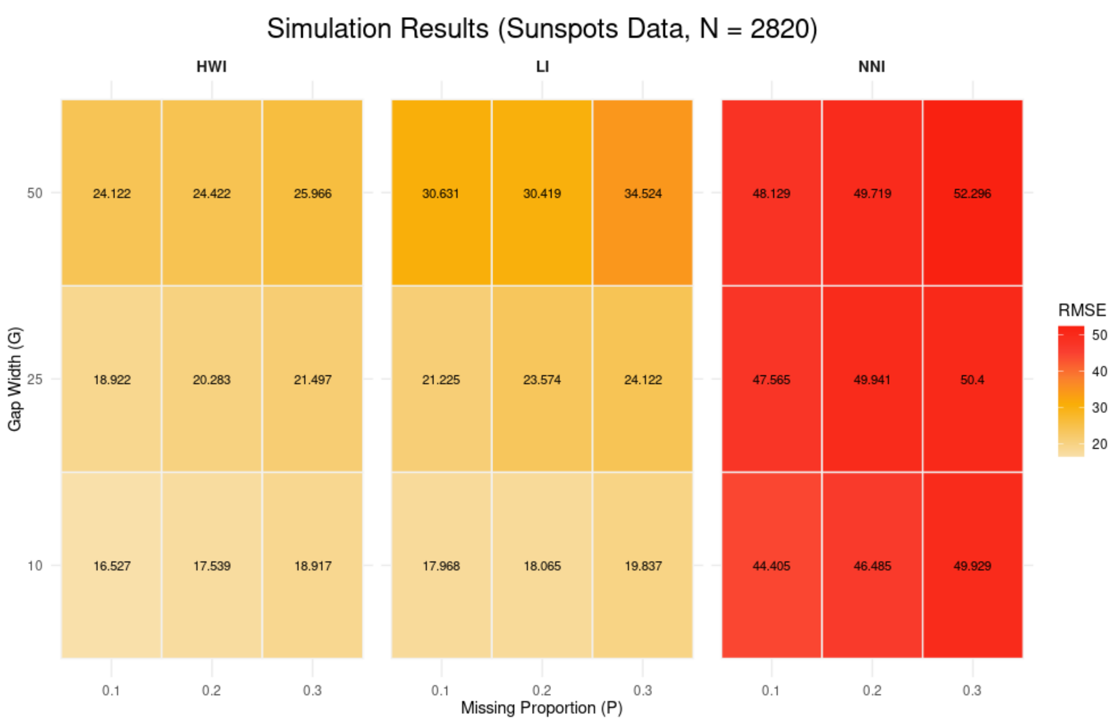


Figure 3: Sunspots data RMSE performance

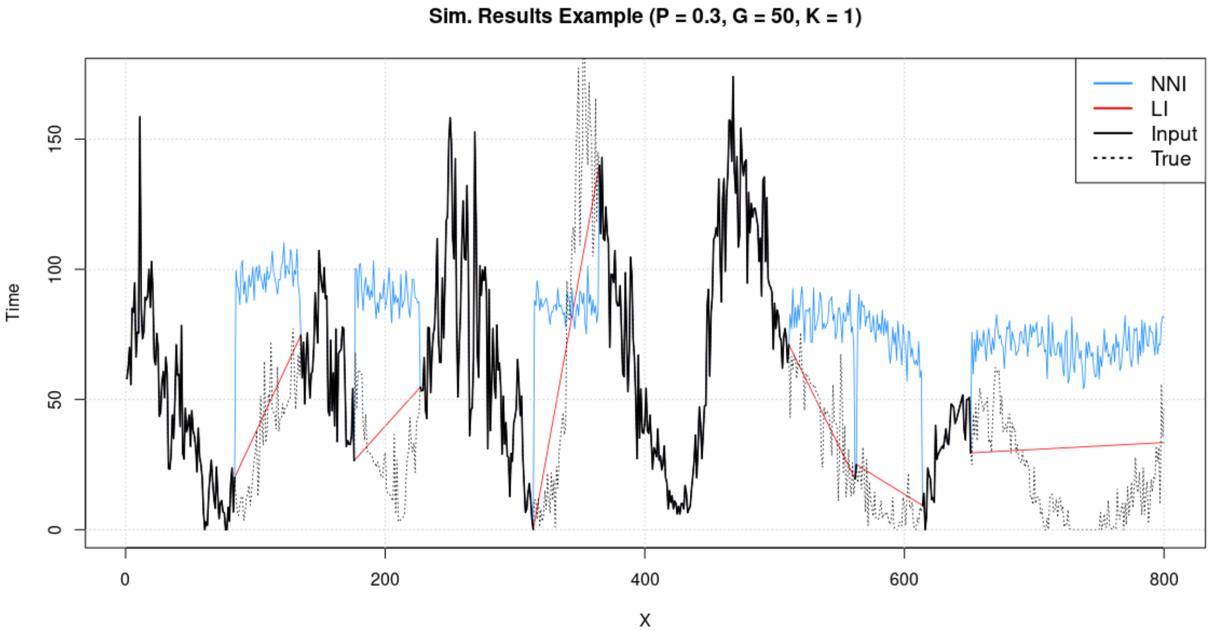
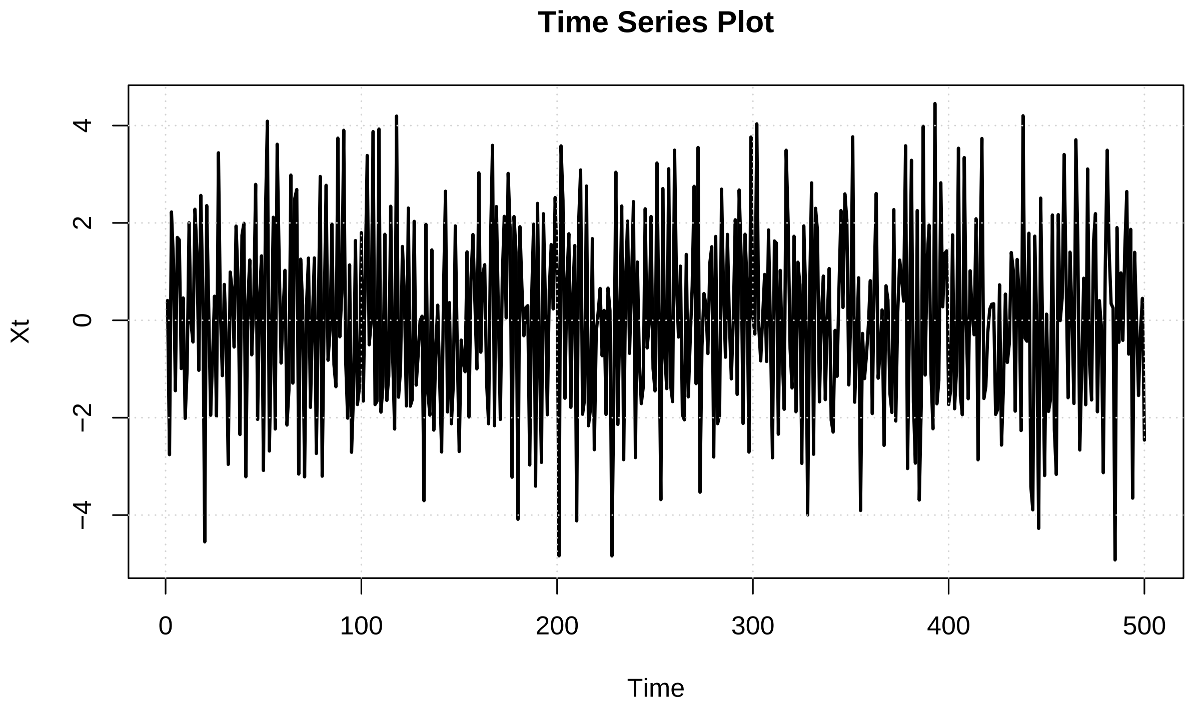


Figure 4: Sunspots performance example

Data Set 4:



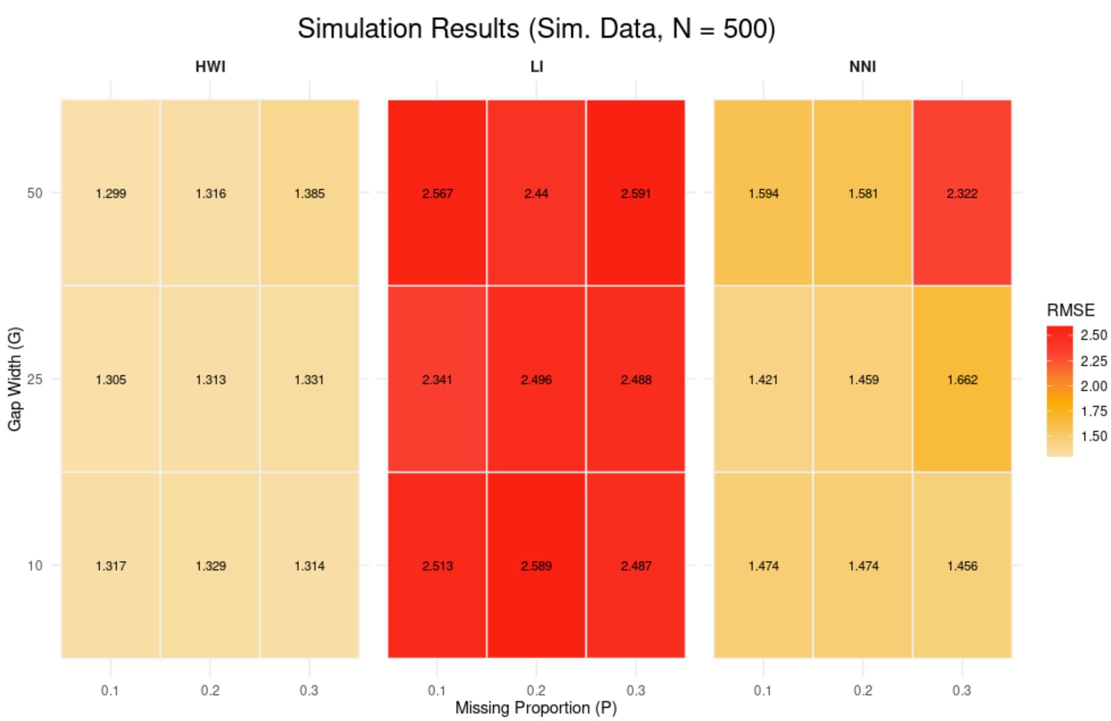
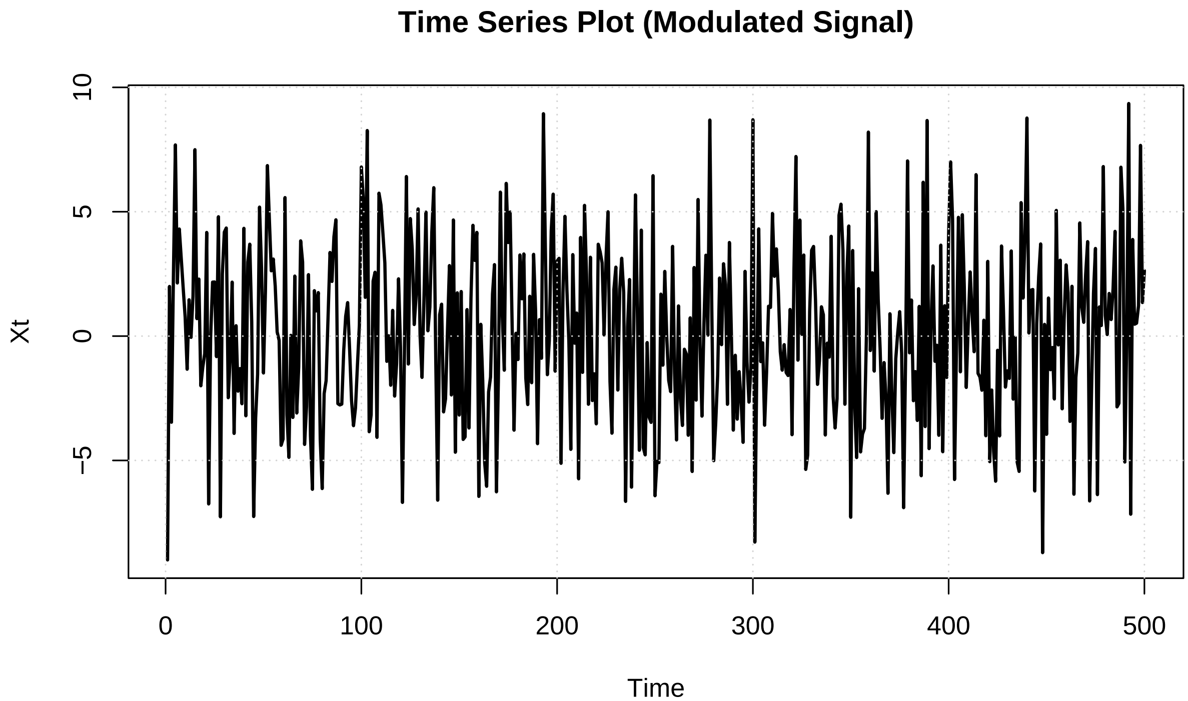


Figure 5: Simulated data RMSE performance

Data Set 5:



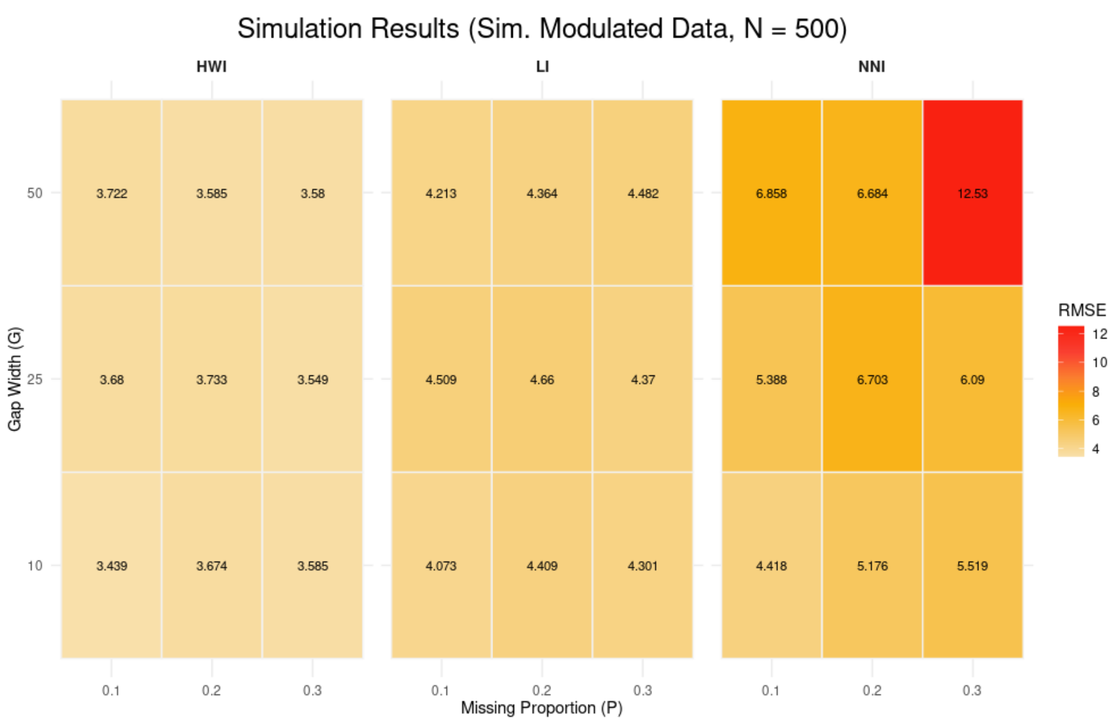


Figure 6: Simulated modulated data RMSE performance