Time series analysis in the frequency domain

FISH 550 – Applied Time Series Analysis

Mark Scheuerell

11 May 2023

Topics for today

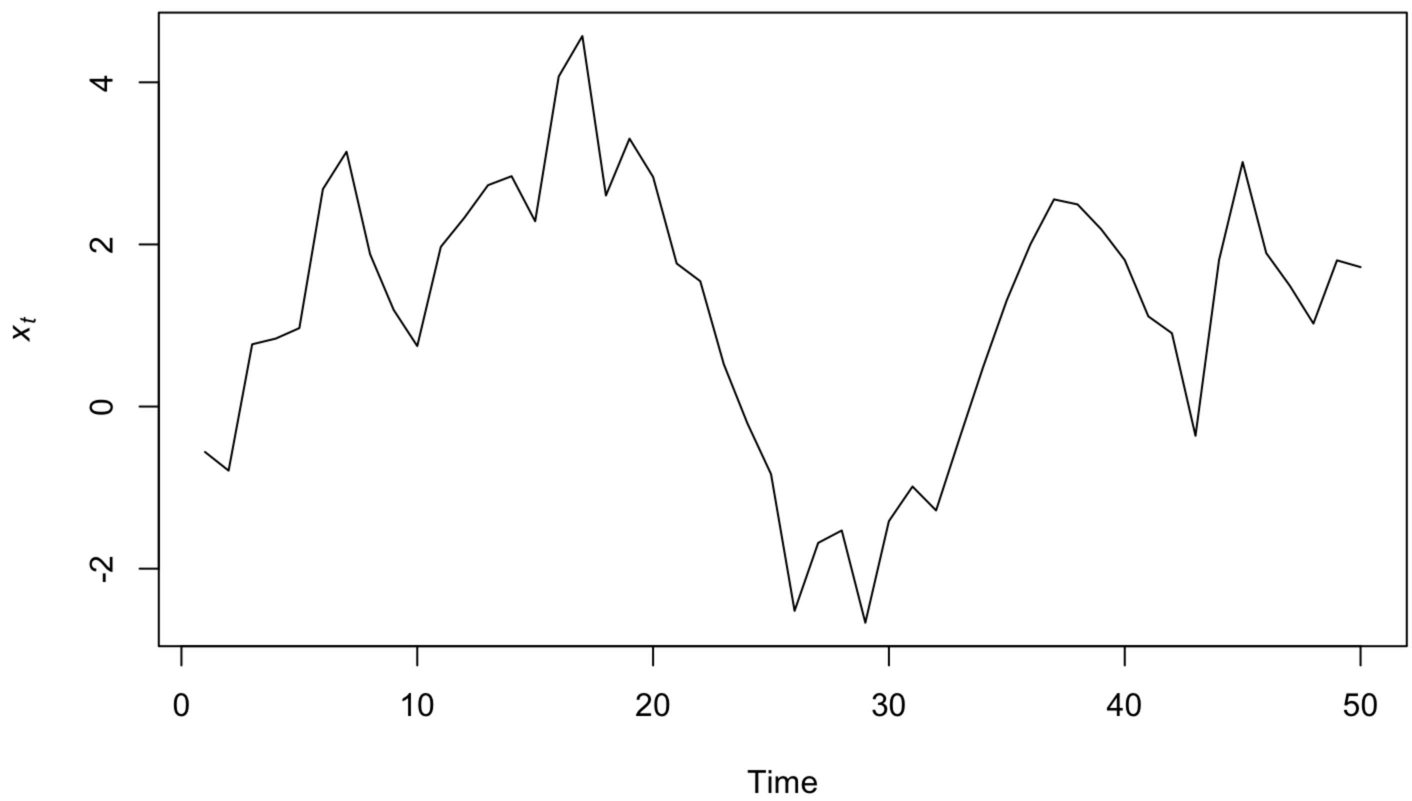
What is the frequency domain?

Fourier transforms

Spectral analysis

Wavelets

Time domain



We having been examining changes in x_t over time

Time domain

We can think of this as comparing changes in amplitude (displacement) with time

Frequency domain

Today we'll consider how amplitude changes with frequency

Jean-Baptiste Fourier (1768 - 1830)

French mathematician & physicist best known for his studies of heat transfer

First described what we now call the “greenhouse effect”

Solving hard problems

Solving the heat equation involves solving *partial differential equations* conditional on some boundary conditions

$$\text{Problem} \Big\downarrow \textit{really hard} \; \text{Solution}$$

Fourier's approach

Find $f(t)$ and $\hat{f}(t)$, such that

$$
\begin{tikzcd}
\text{Problem} \arrow[r, "f(t)"] \arrow[d, "really"', "hard"] & \text{Transformed problem} \arrow[d, "much"', "easier"] \\
\text{Solution} & \text{Transformed solution} \arrow[l, "\hat{f}(t)"']
\end{tikzcd}
$$

Fourier series

Complex periodic functions can be written as infinite sums of sine waves

$$f(t) = a_0 + \sum_{k=1}^{\infty} a_k \sin(2\pi f_0 kt + p_k)$$

where

k is the wave number (index)

a_k is the amplitude of wave k

f_0 is the fundamental frequency

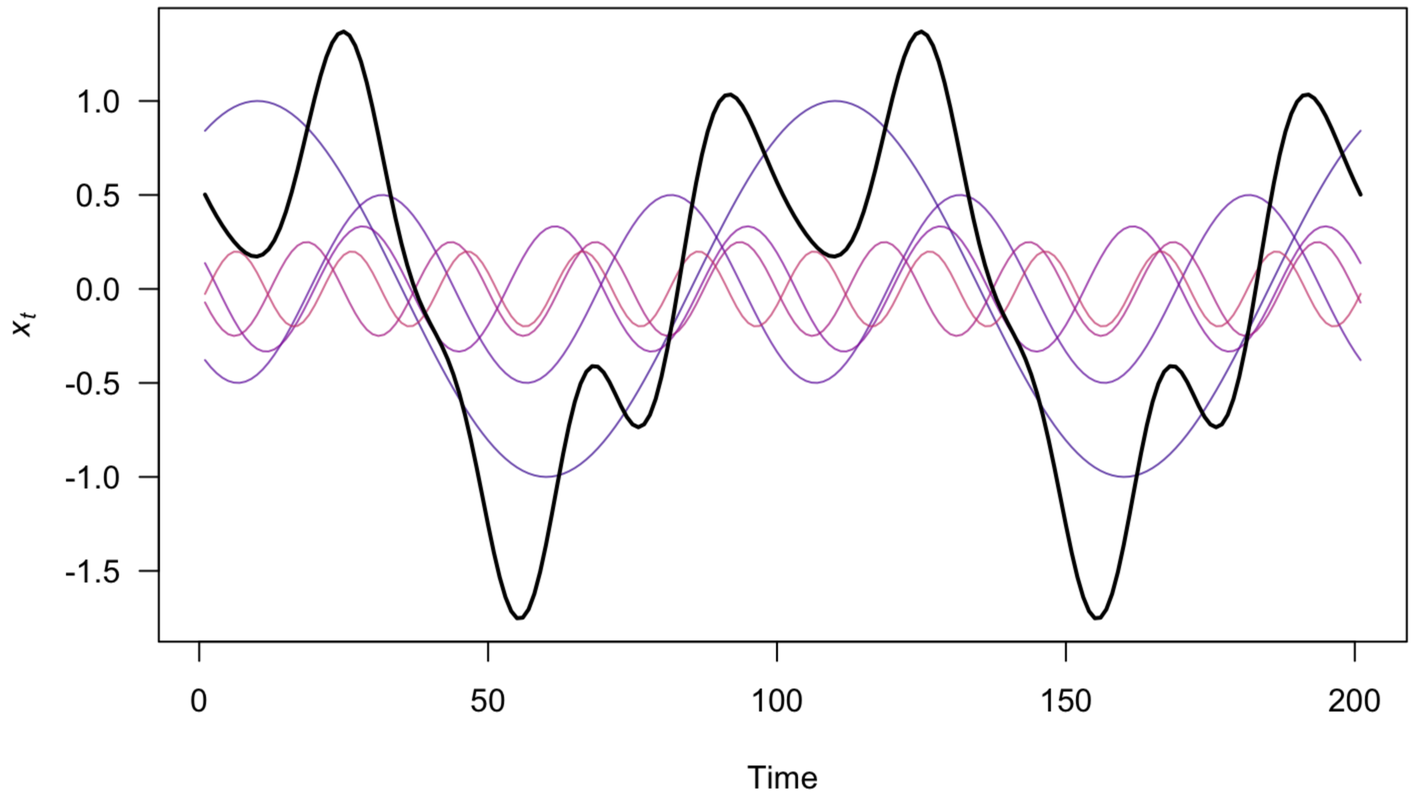
p_k is the phase shift

Fourier series

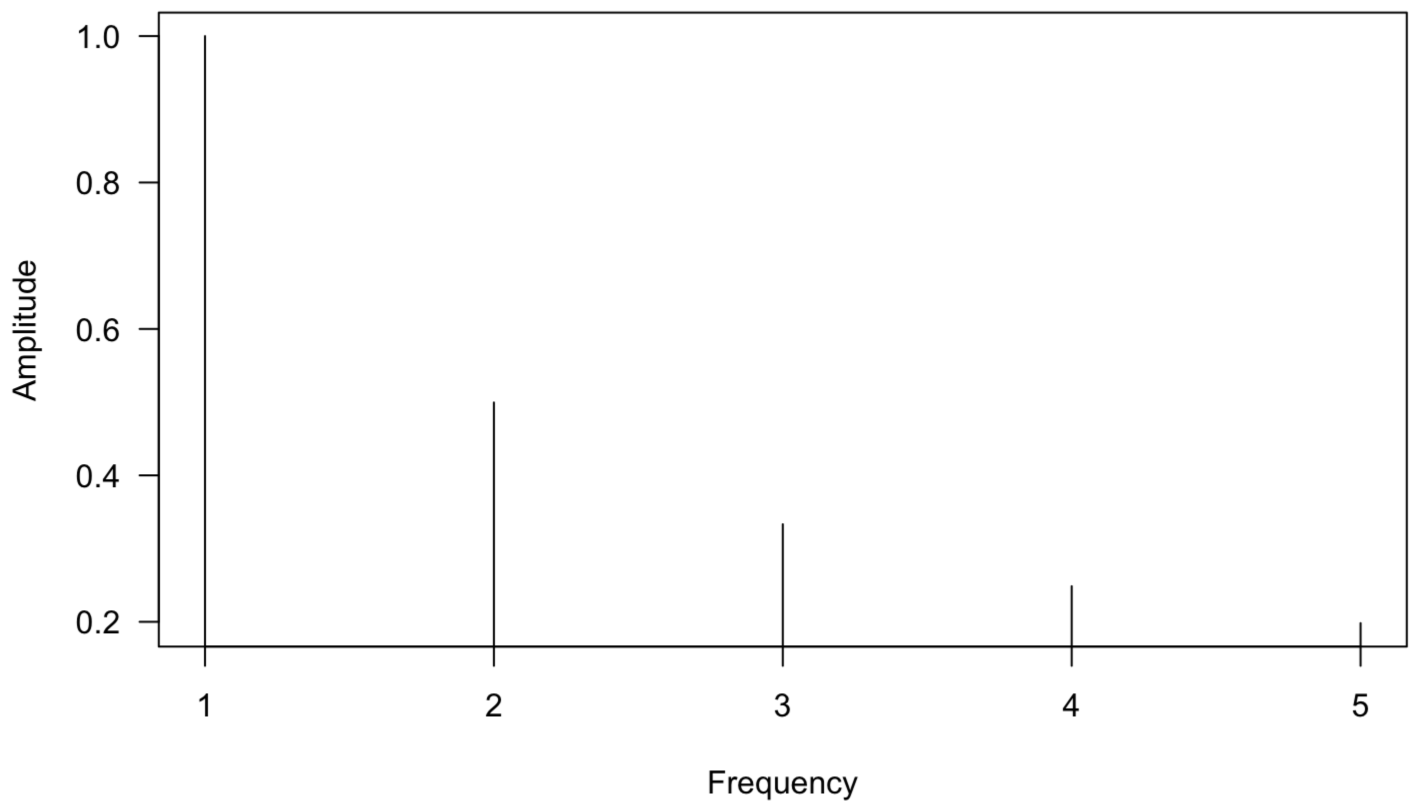
A finite example

$$f(t) = \sum_{k=1}^{5} \frac{1}{k} \sin(2\pi k t + k^2)$$

Fourier series



Fourier series



Fourier series

Here's an animated example from Wikipedia

Fourier transform

We can make use of Euler's formula

$$\cos(2\pi k) + i\sin(2\pi k) = e^{i2\pi k}$$

and write the Fourier transform of $f(t)$ as

$$f(t) = \int_{-\infty}^{\infty} \hat{f}(k)\, e^{i2\pi tk}\, dk$$

where k is the frequency

Discrete Fourier transform

Fourier transform

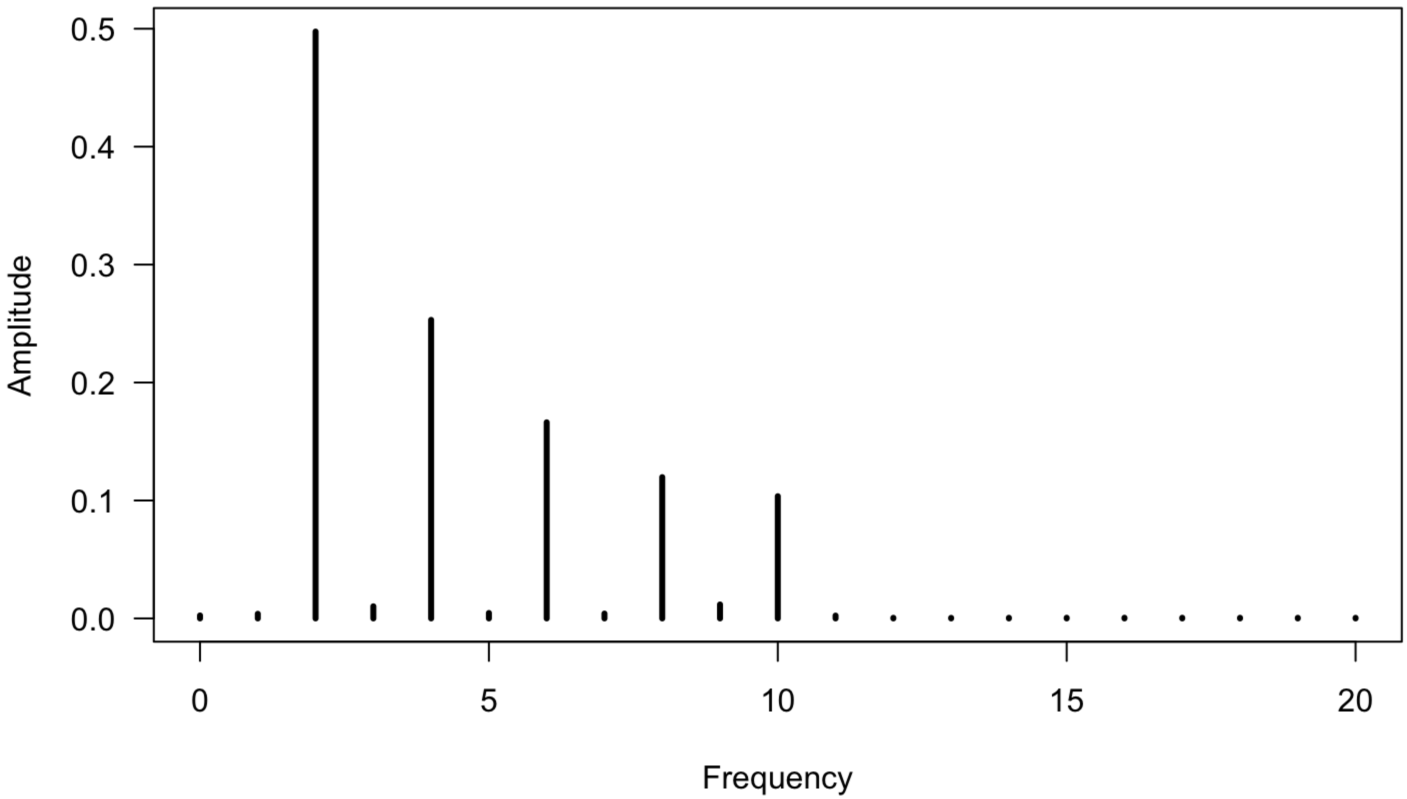
$$f_k = \sum_{n=0}^{N-1} x_t \, e^{-i2\pi nk}$$

Fourier transforms in R

R uses what's known as *Fast Fourier transform* via `fft()`, which returns the amplitude at each frequency

```
ft <- fft(xt)
## often normalize by the length
ft <- fft(xt) / length(xt)
```

Fourier represention of our $\{x_t\}$



Discrete Inverse Fourier transform

Fourier transform

$$f_k = \sum_{n=0}^{N-1} x_t \; e^{-i2\pi nk}$$

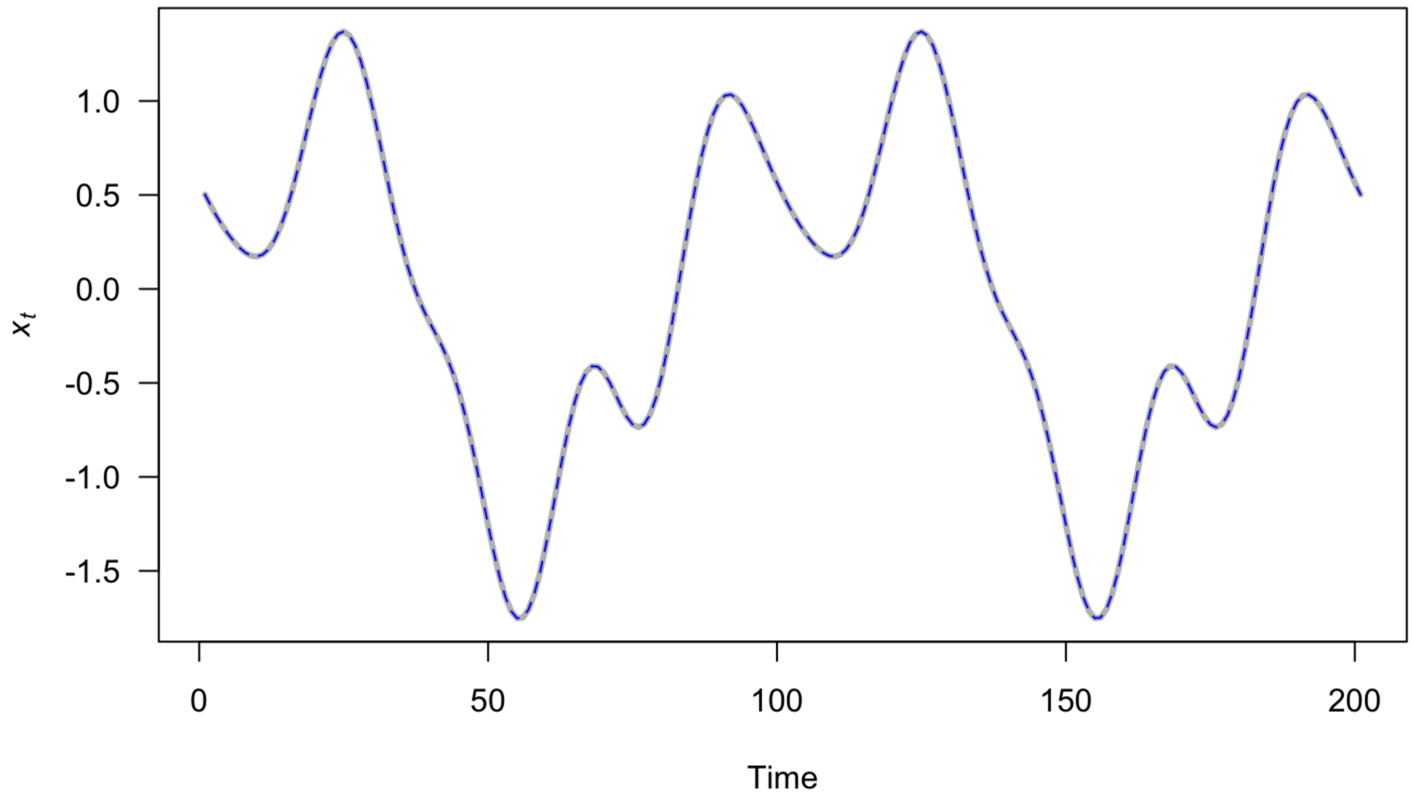
Inverse

$$x_t = \sum_{k=0}^{N-1} f_k \; e^{i2\pi nk}$$

Inverse Fourier transforms in R

```
 i <- complex(1, re = 0, im = 1)
xx <- rep(NA, TT)
kk <- seq(TT) - 1
## Inverse Fourier transform
## ft <- fft(xt)
for(t in kk) {
  xx[t+1] <- sum(ft * exp(i*2*pi*kk*t/TT))
}
```

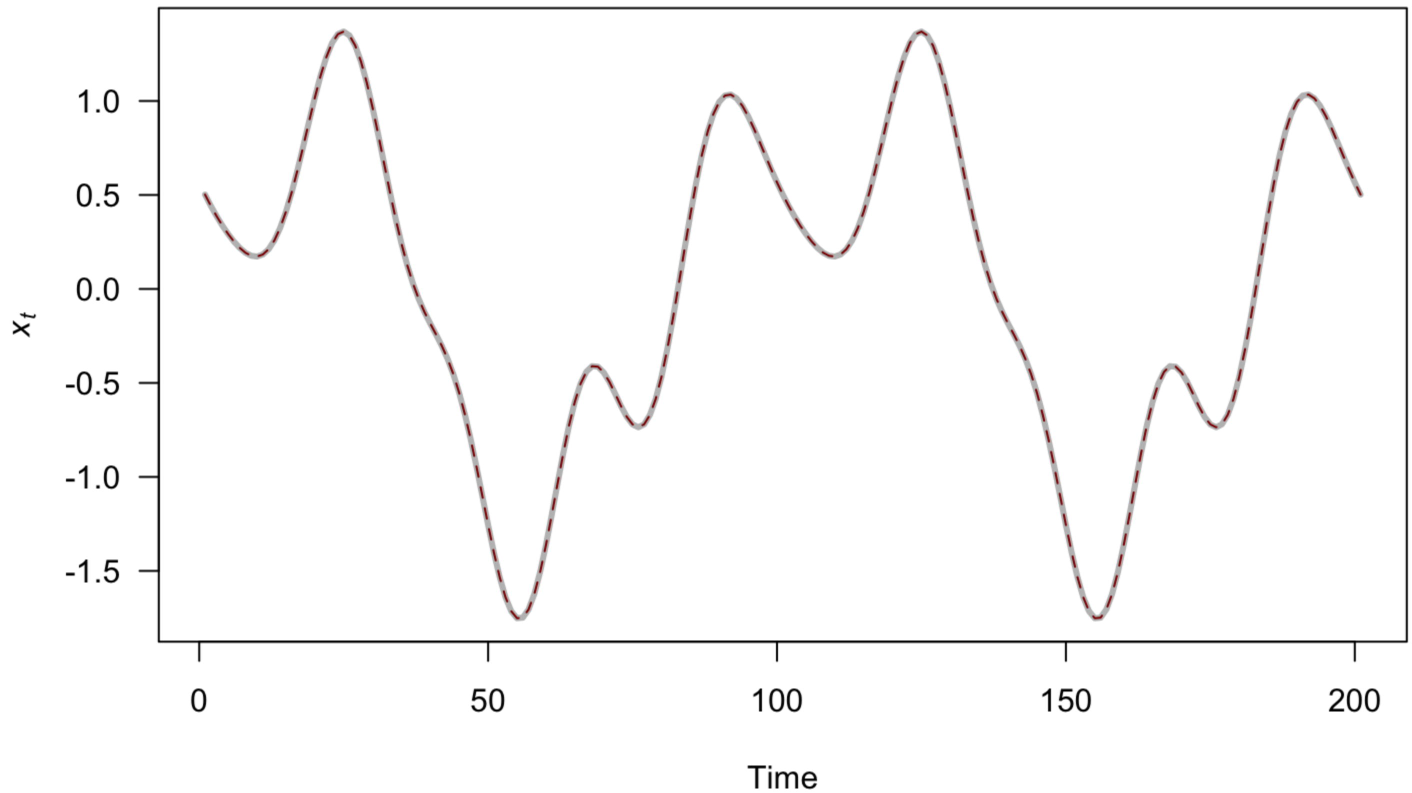
Original $\{ x_t \}$ & our inverse transform



Inverse Fourier transforms in R

```
ift <- fft(ft, inverse = TRUE)
```

Original $\{x_t\}$ & R's inverse transform



Spectral analysis

Spectral analysis

Spectral analysis refers to a *general* way of decomposing time series into their constituent frequencies

Spectral analysis

Consider a linear regression model for $\{x_t\}$ with various sines and cosines as predictors

$$x_t = a_0 + \sum_{k=1}^{n/2-1} a_k \cos(2\pi f_0 kt/n) + b_k \sin(2\pi f_0 kt/n)$$

Periodogram

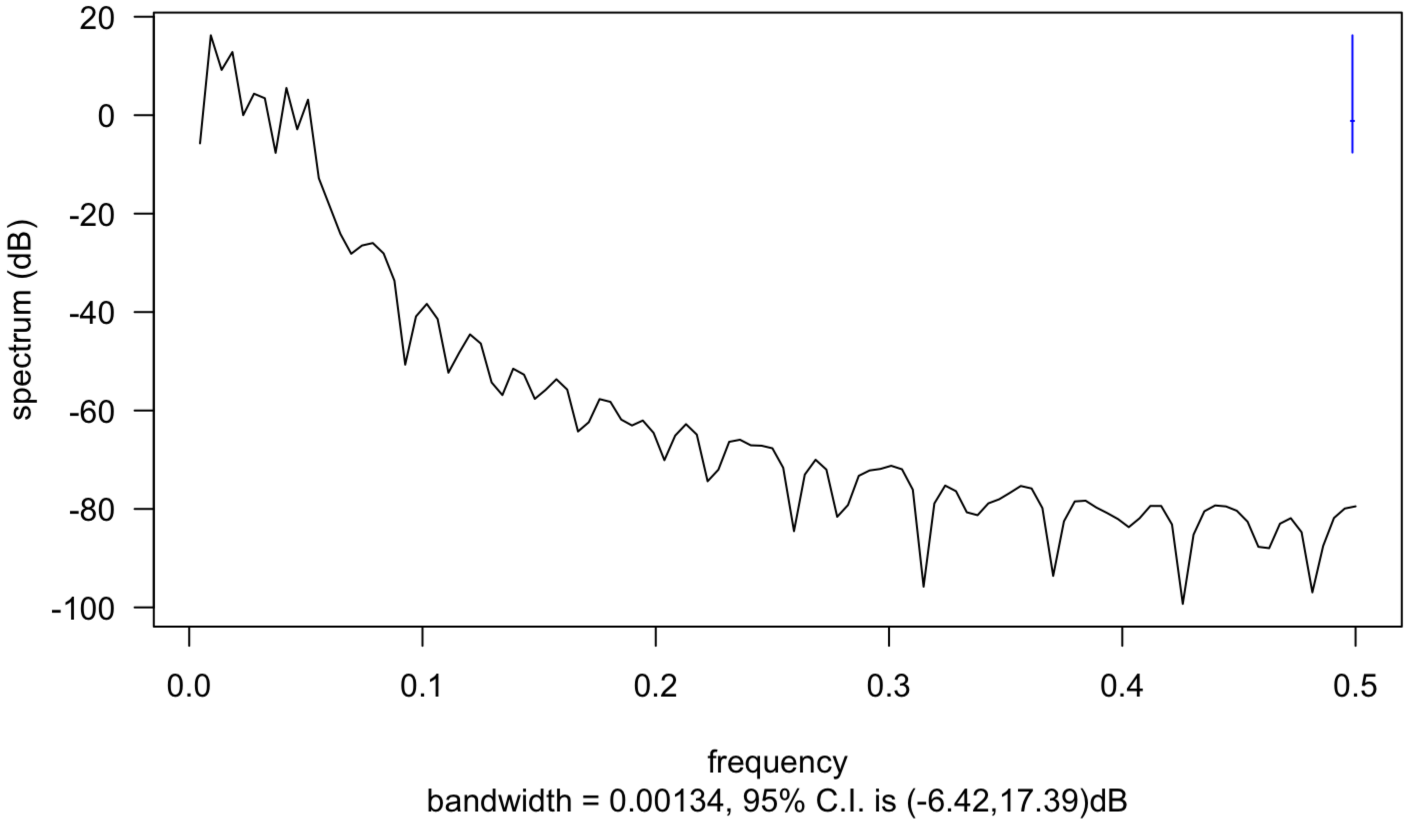
The *periodogram* measures the contributions of each frequency k to $\{x_t\}$

$$P_k = a_k^2 + b_k^2$$

Estimate the periodogram in R

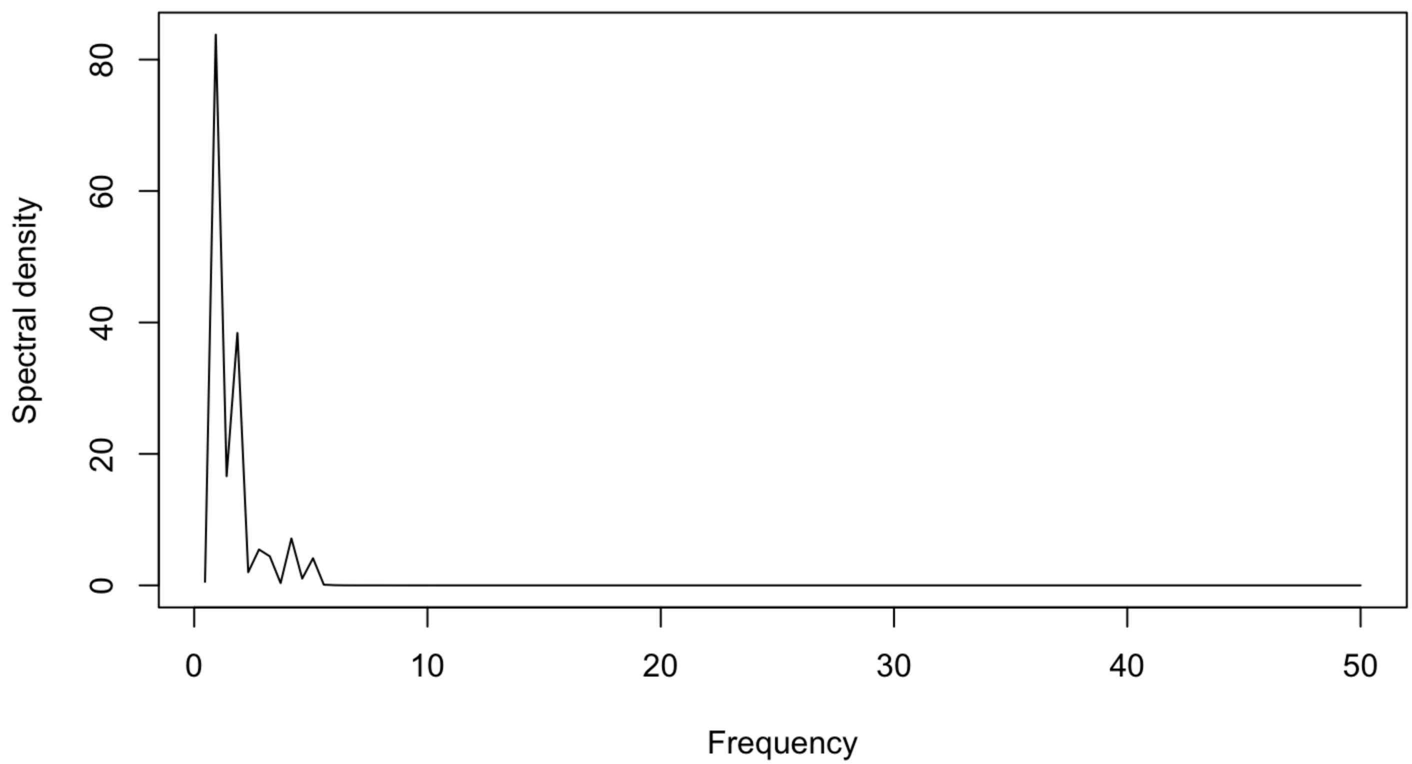
```r
spectrum(xt, log = "on")
spectrum(xt, log = "off")
spectrum(xt, log = "dB")
```

Periodogram for our $\{x_t\}$




```
spectrum(xt, log = "dB")
```

Periodogram for our $\{x_t\}$



Density on natural scale & frequency in cycles per time

Spectral density estimation via AR(p)

For an AR(p) process

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots + \phi_p x_{t-p} + e_t$$

The spectral density is

$$S(f, \phi_1, \ldots, \phi_p, \sigma^2) = \frac{\sigma^2 \Delta t}{|1 - \sum_{k=1}^{p} \phi_k e^{-i2\pi f k \Delta t}|^2}$$

Limits to spectral analysis

Spectral analysis works well for

1. stationary time series
2. identifying periodic signals corrupted by noise

Limits to spectral analysis

Spectral analysis works well for

1. stationary time series
2. identifying periodic signals corrupted by noise

But…

1. it's an inconsistent estimator for most real data sets
2. it's generally biased

Wavelets

Shifting frequencies

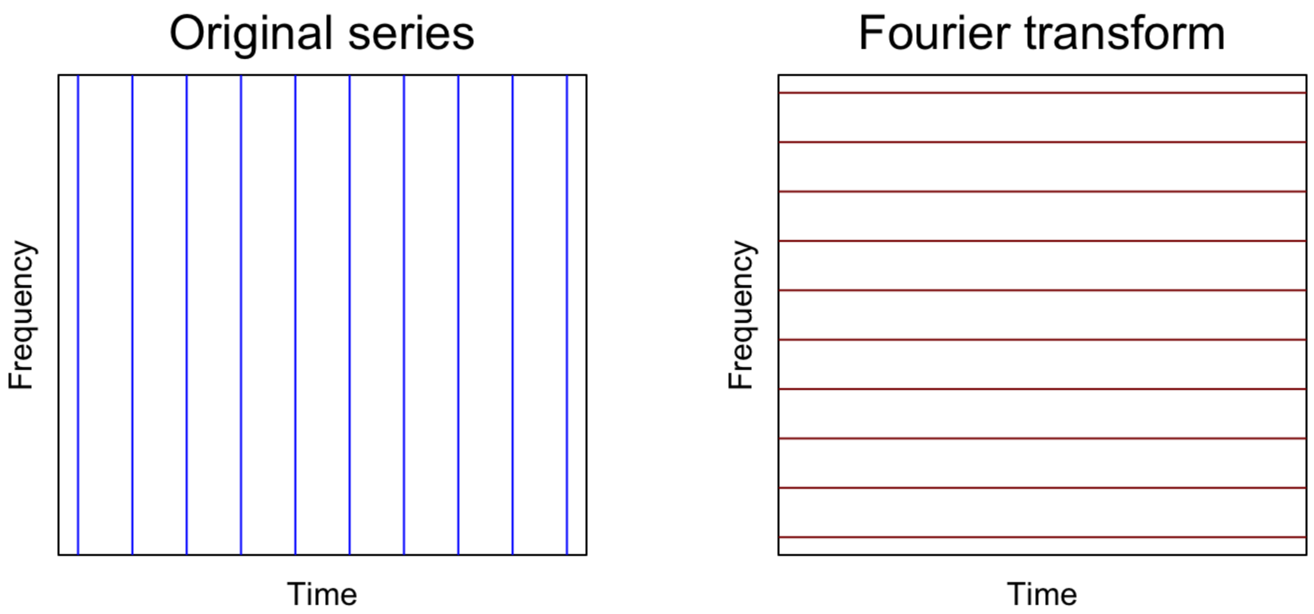
What if the frequency changes over time?

Wavelets

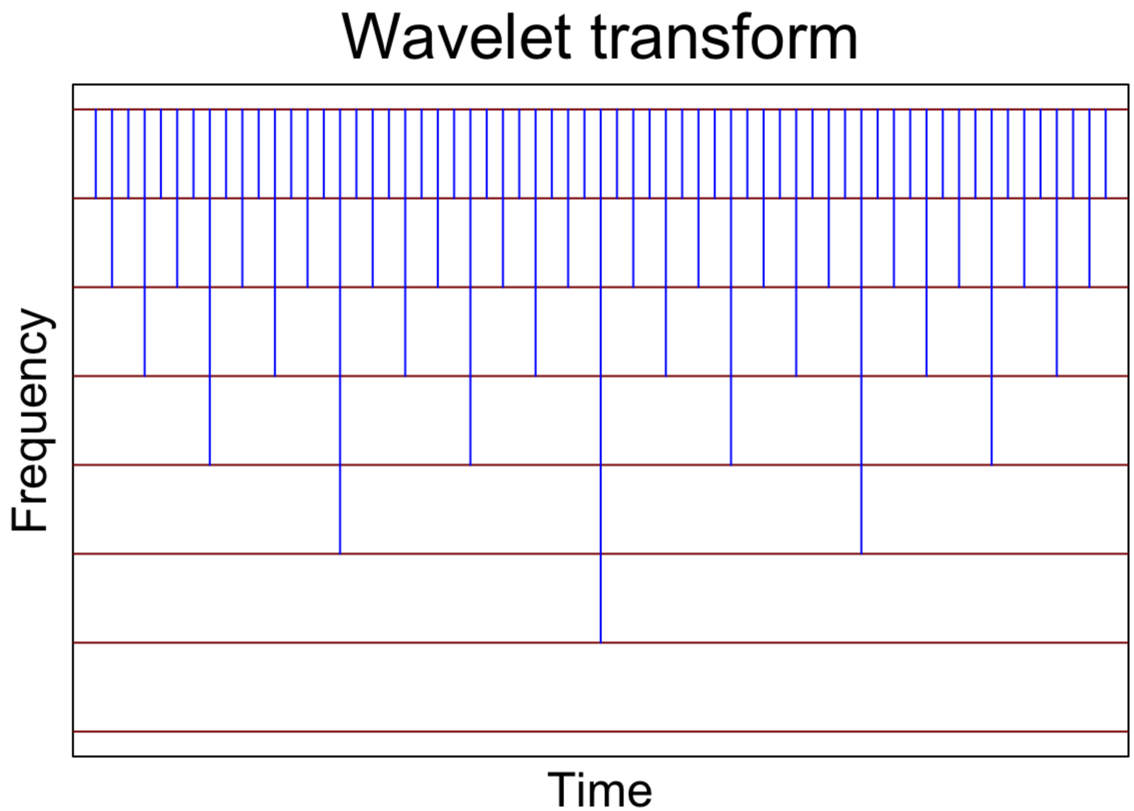
For non-stationary time series we can use so-called *wavelets*

A wavelet is a function that is localized in time & frequency

Graphical forms for decomposition



Graphical form for decomposition



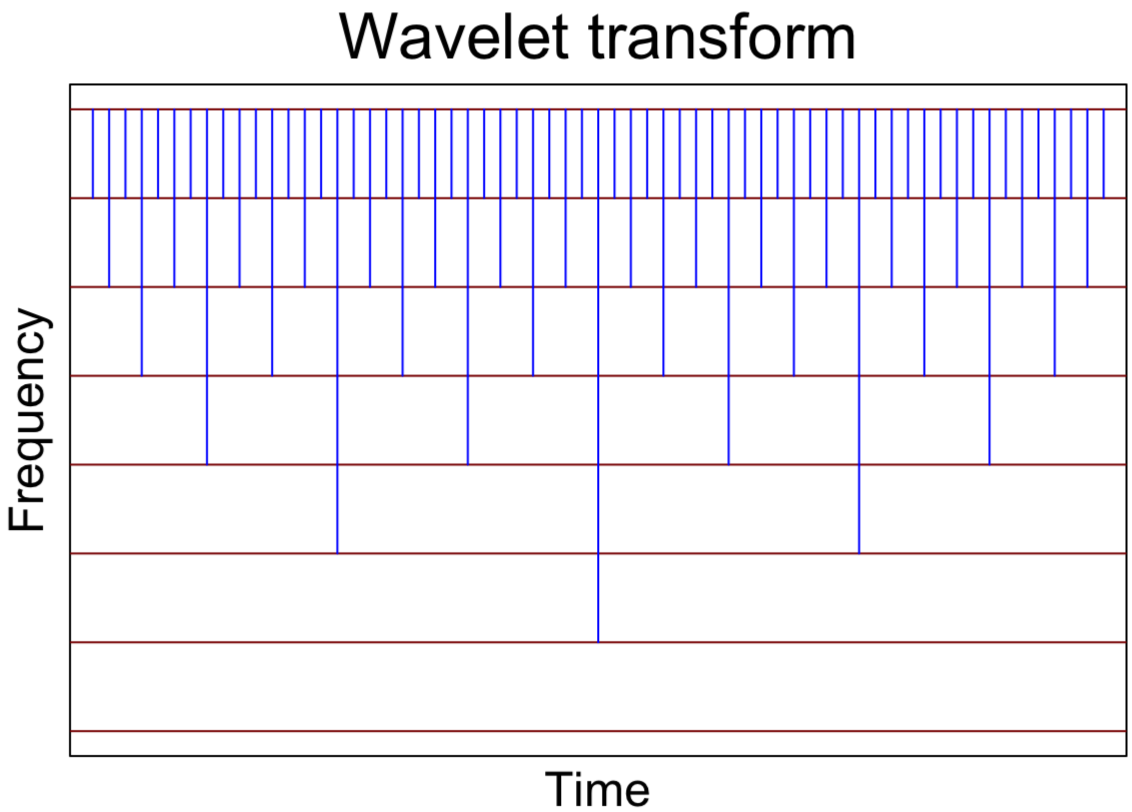
What is a wavelet?

Formally, a wavelet ψ is defined as

$$\psi_{\sigma,\tau}(t) = \frac{1}{\sqrt{|\sigma|}}\psi\left(\frac{t-\tau}{\sigma}\right)$$

where τ determines its position & σ determines its frequency

Graphical form for decomposition



Properties of wavelets

It goes up **and** down

$$\int_{-\infty}^{\infty} \psi(t)\, dt = 0$$

It has a finite sum

$$\int_{-\infty}^{\infty} |\psi(t)|\, dt < \infty$$

How are wavelets defined?

In terms of scaling functions that describe

1. Dilations $\psi(t) \rightarrow \psi(2t)$
2. Translations $\psi(t) \rightarrow \psi(t-1)$

How are wavelets defined?

More generally,

$$\psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k)$$

where

j is the dilation index

k is the translation index

and

$2^{j/2}$ is a normalization constant

Wavelets in practice

There are many options for $\psi(t)$, but we'll use scaling functions and define

$$\psi(t) = \sum_{k=0}^{K} c_k \psi(2x - k)$$

where the c_k are filter coefficients*

*Note that $\psi(t)$ gets "smoother" as K increases

Haar's scaling function

Simple, but commonly used, where $K = 1;\ c_0 = 1;\ c_1 = 1$

$$\psi(t) = \sum_{k=0}^{K} c_k \psi(2t - k)$$

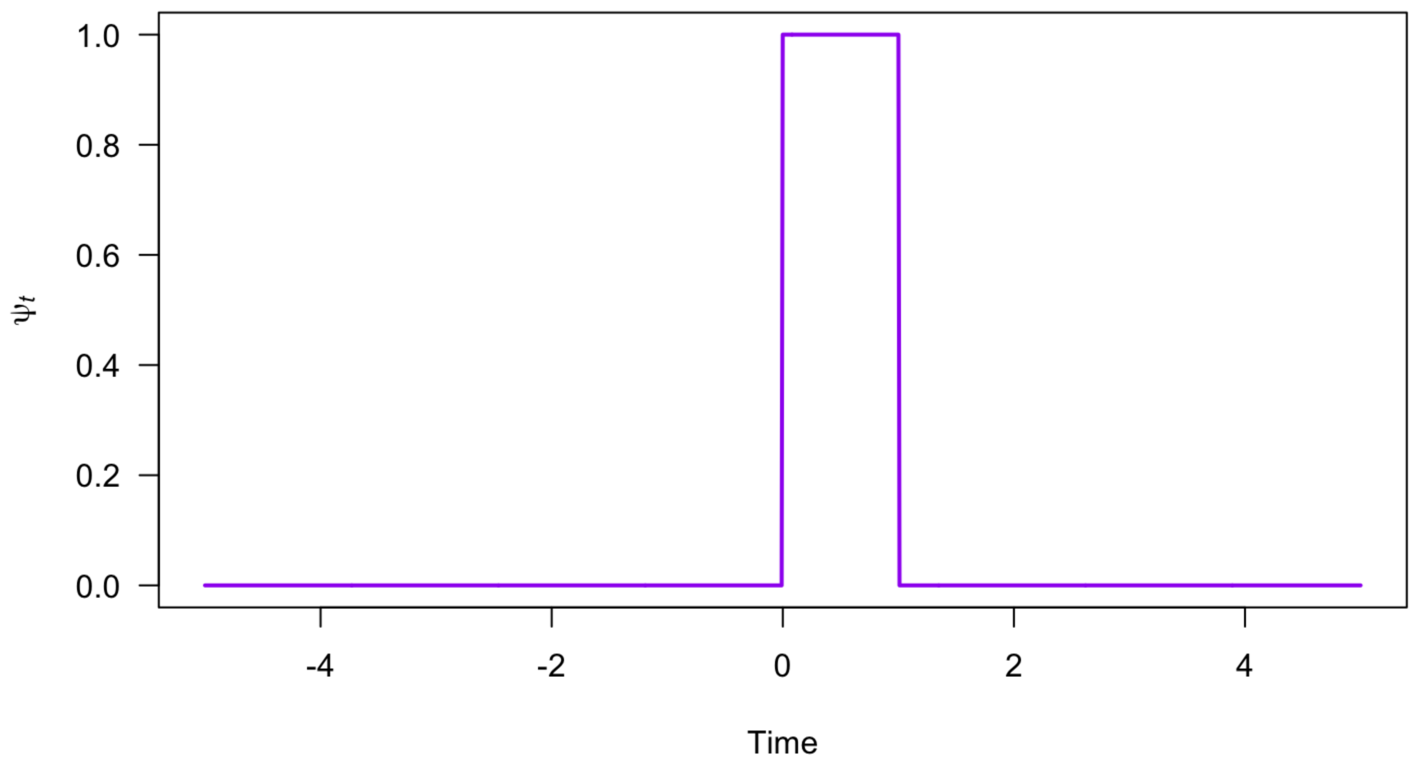
$$\Downarrow$$

$$\psi(t) = \psi(2t) + \psi(2t - 1)$$

The only function that satisfies this is:

$$\psi(t) = 1 \text{ if } 0 \leq t \leq 1$$
$$\psi(t) = 0 \text{ otherwise}$$

Haar's scaling function



Haar's scaling function

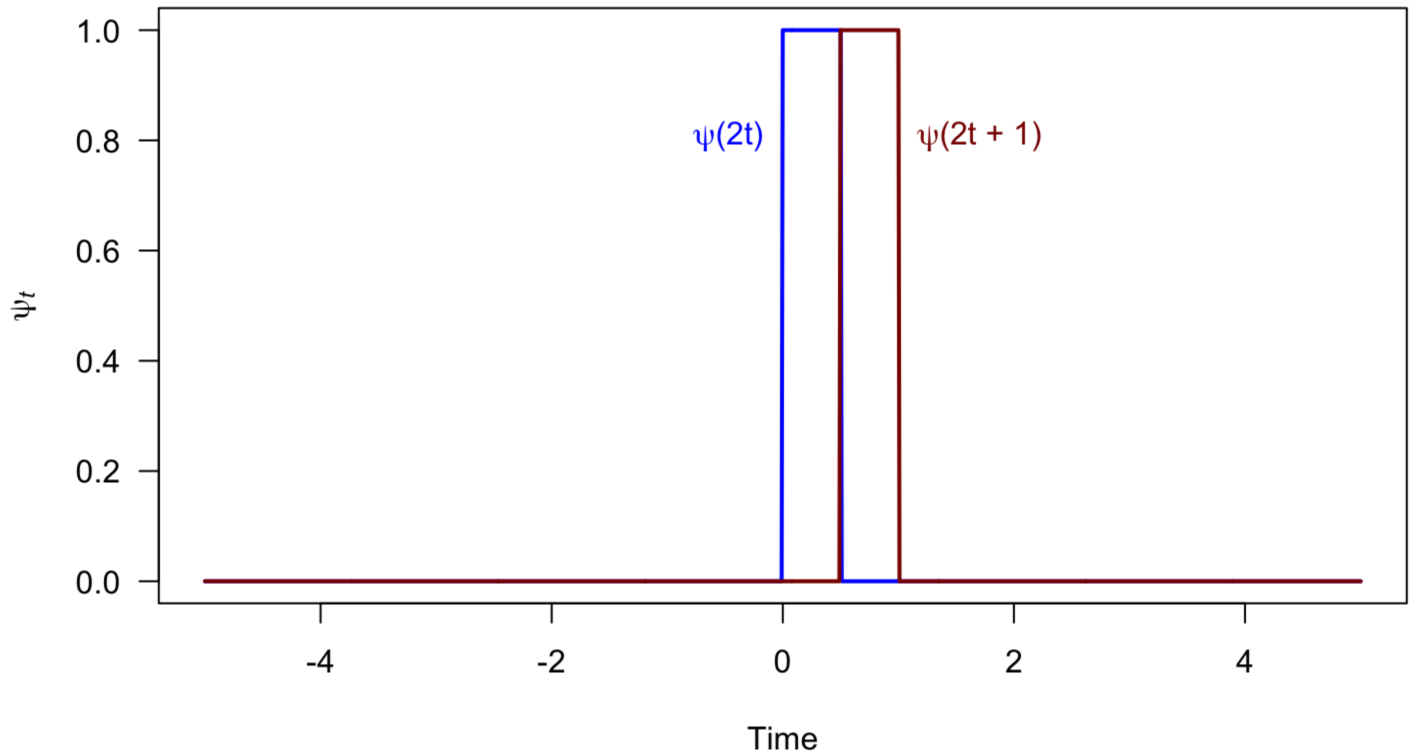
In terms of the dilation

$$\psi(2t) = 1 \text{ if } 0 \leq t \leq 0.5$$
$$\psi(2t) = 0 \text{ otherwise}$$

and translation

$$\psi(2t-1) = 1 \text{ if } 0.5 \leq t \leq 1$$
$$\psi(2t-1) = 0 \text{ otherwise}$$

Haar's scaling function (father)



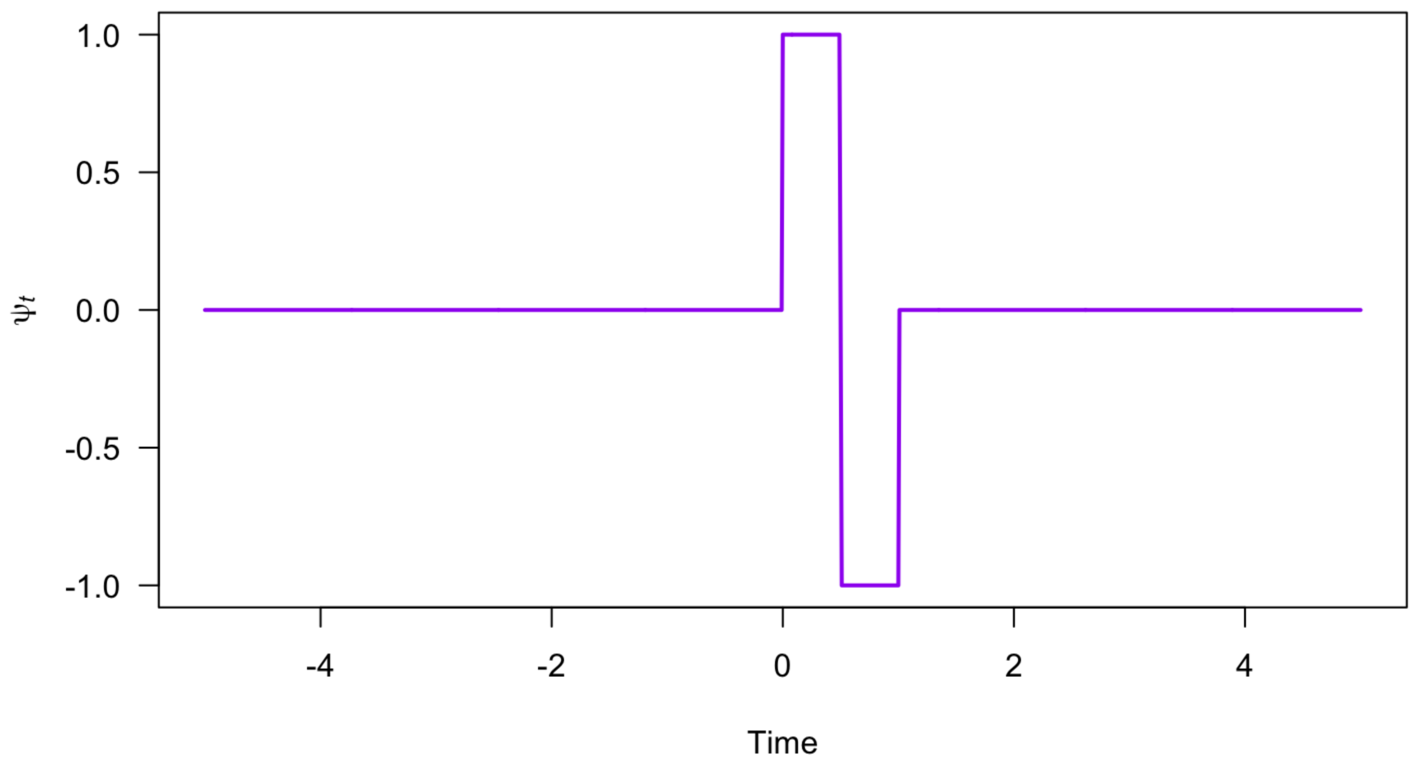
Haar's mother wavelet

Wavelets are created via differencing of scaling functions

$$\psi(t) = \sum_{k=0}^{1} (-1)^k c_k \psi(2t - k)$$

where $(-1)^k$ creates the difference

Haar's mother wavelet



Family of Haar's wavelets

So-called "child" wavelets are created via dilation & translation

$$\psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k)$$

The mother Haar wavelet has $j = 0$

Family of Haar's wavelets

So-called "child" wavelets are created via dilation & translation

$$\psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k)$$

The basic Haar wavelet has $j = 0$

Setting $j = 1$ yields a daughter

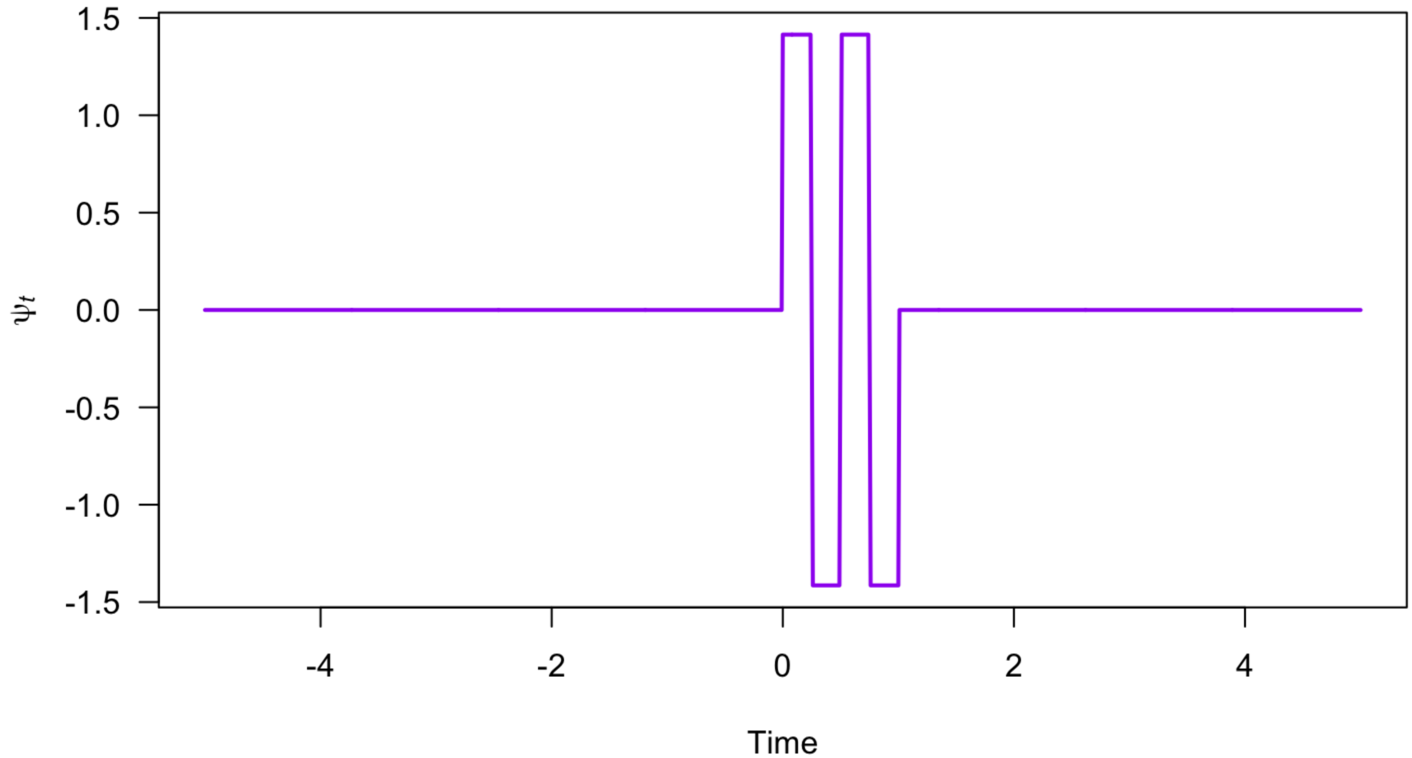
$$\psi_{j,k}(t) = \sqrt{2}\psi(2t - k)$$

Haar's daughter wavelet

$$\psi(t) = \sum_{k=0}^{1} (-1)^k c_k \sqrt{2} \psi(2t - k)$$

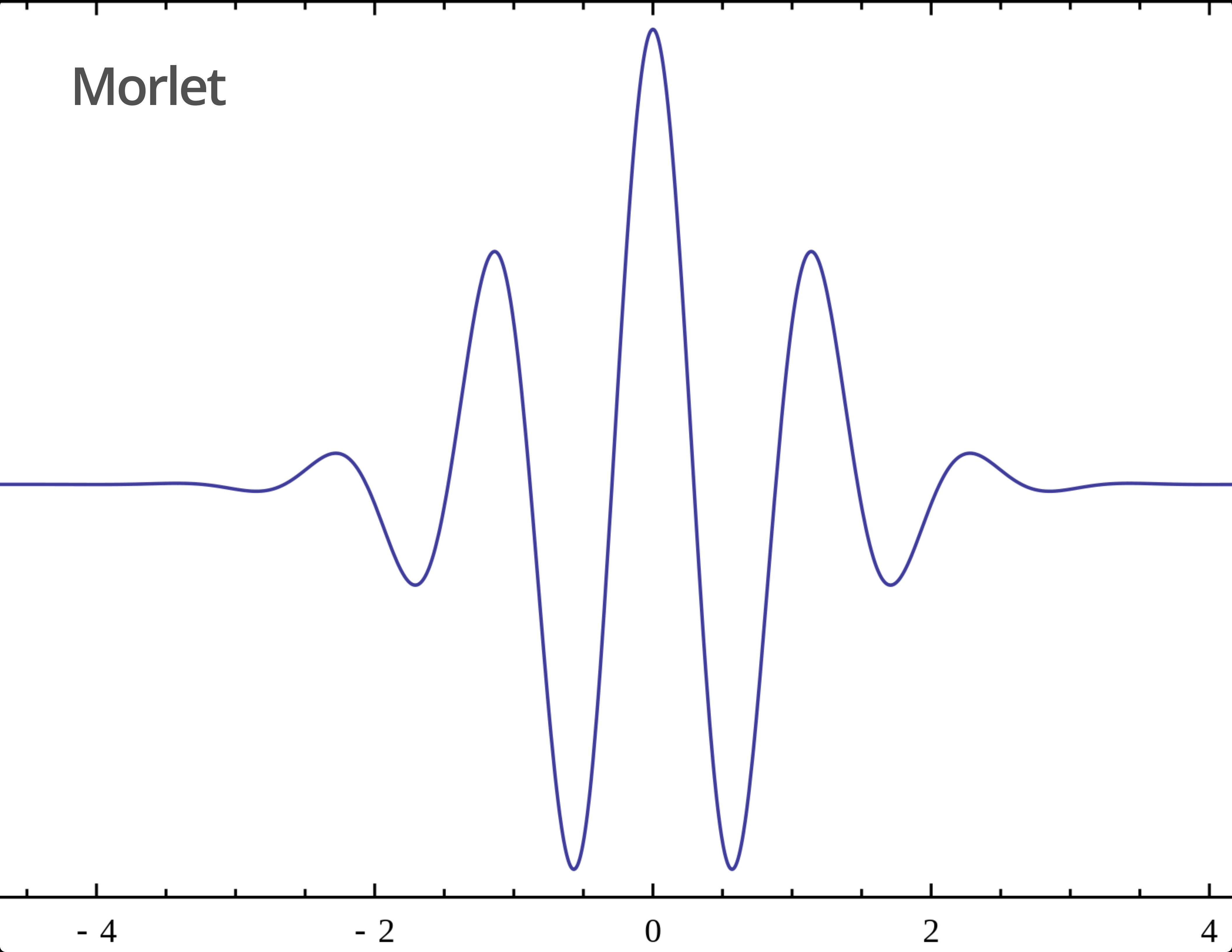
Recall that $(-1)^k$ creates the difference

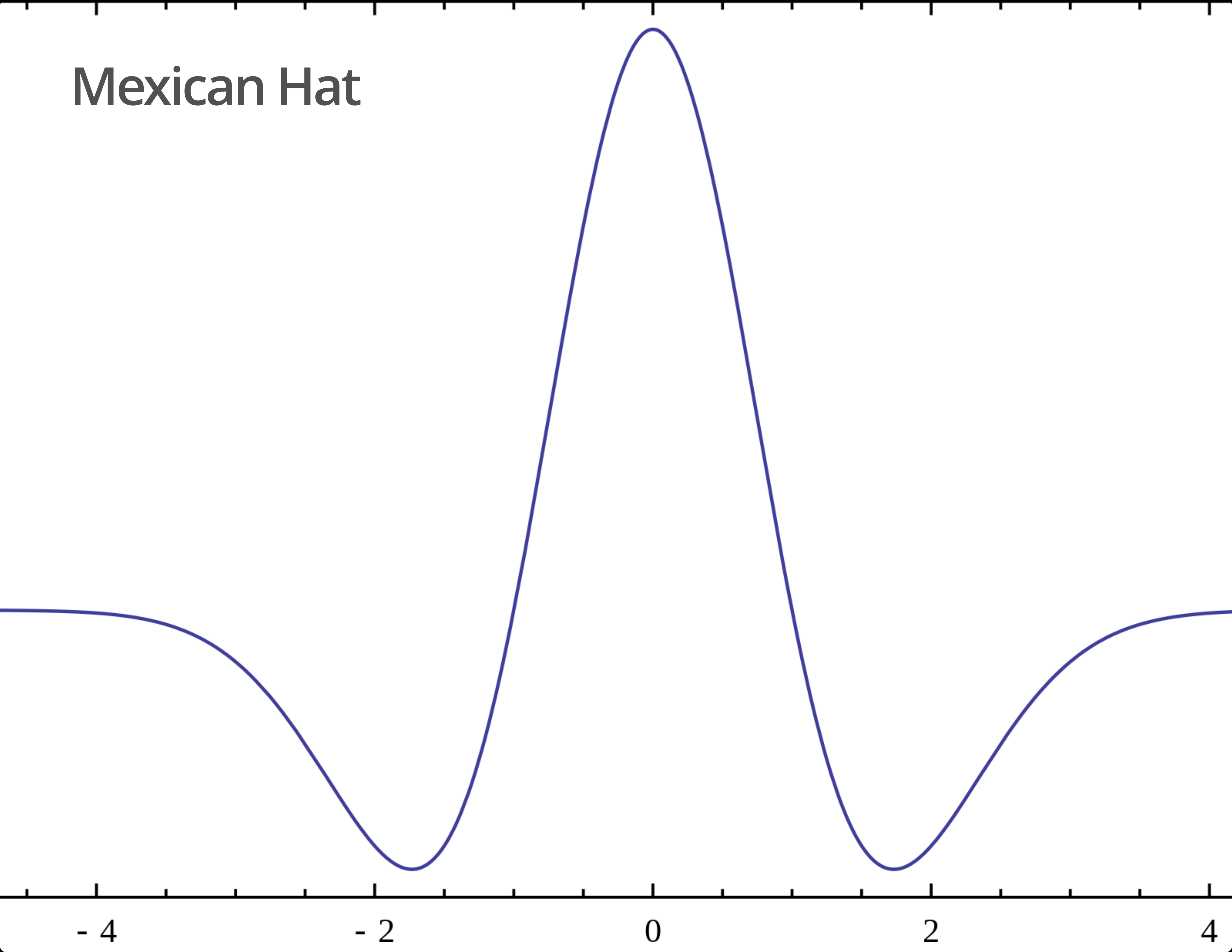
A daughter wavelet of Haar's



Other wavelets

There are many forms of wavelets, many of which were developed in the past 50 years

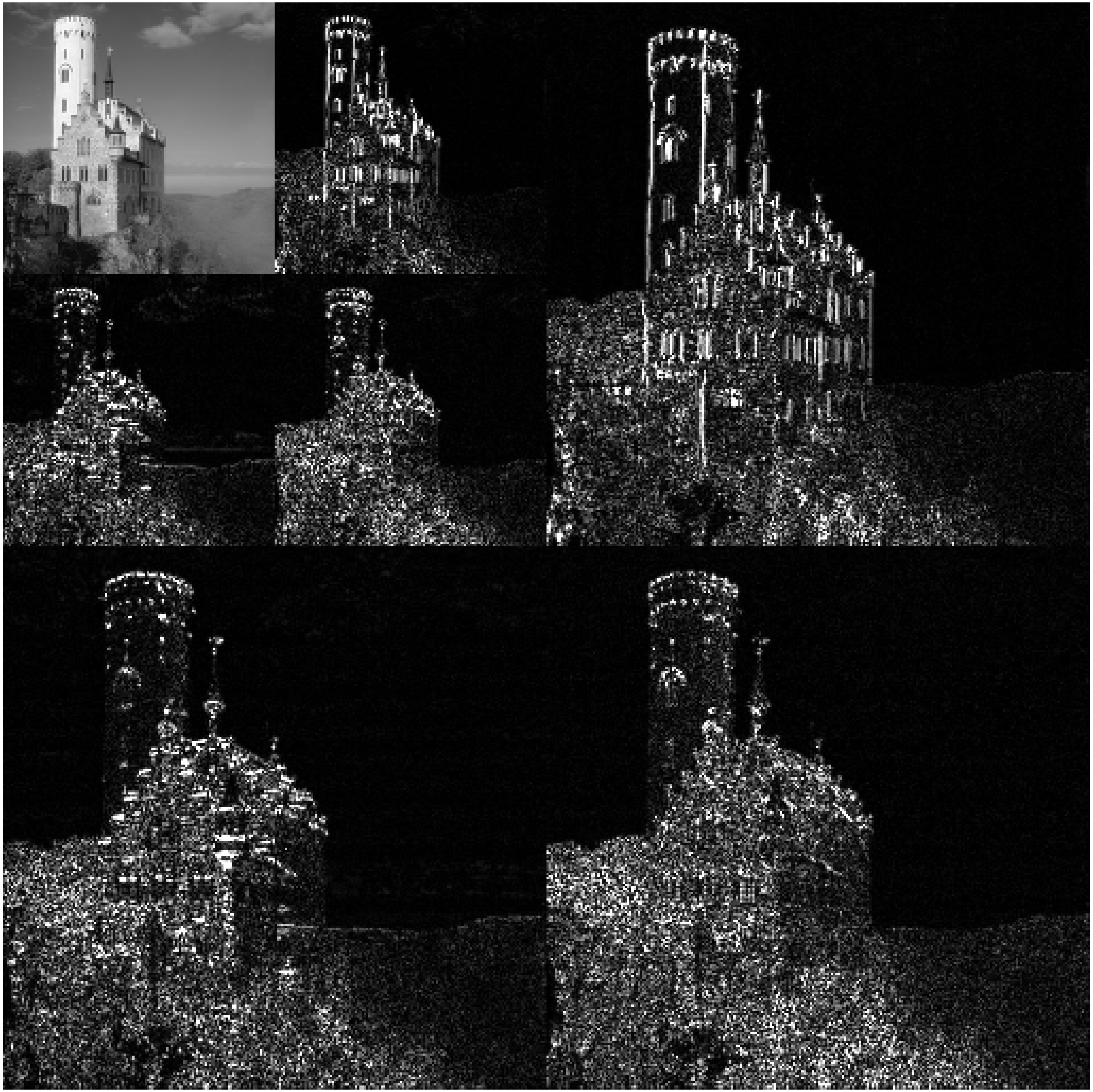

Morlet
- 4
- 2
0
2
4


Mexican Hat
- 4
- 2
0
2
4

Who does this?

Wavelet analysis is used widely in audio & video compression

JPEG



Estimating wavelet transforms in R

We'll use the **WaveletComp** package, which uses the Morlet wavelet

We'll also use the L Washington temperature data from the **MARSS** package

```
library(WaveletComp)
## L WA temperature data
tmp <- MARSS::lakeWAplanktonTrans[,"Temp"]
## WaveletComp needs data as df
dat <- data.frame(tmp = tmp)
```

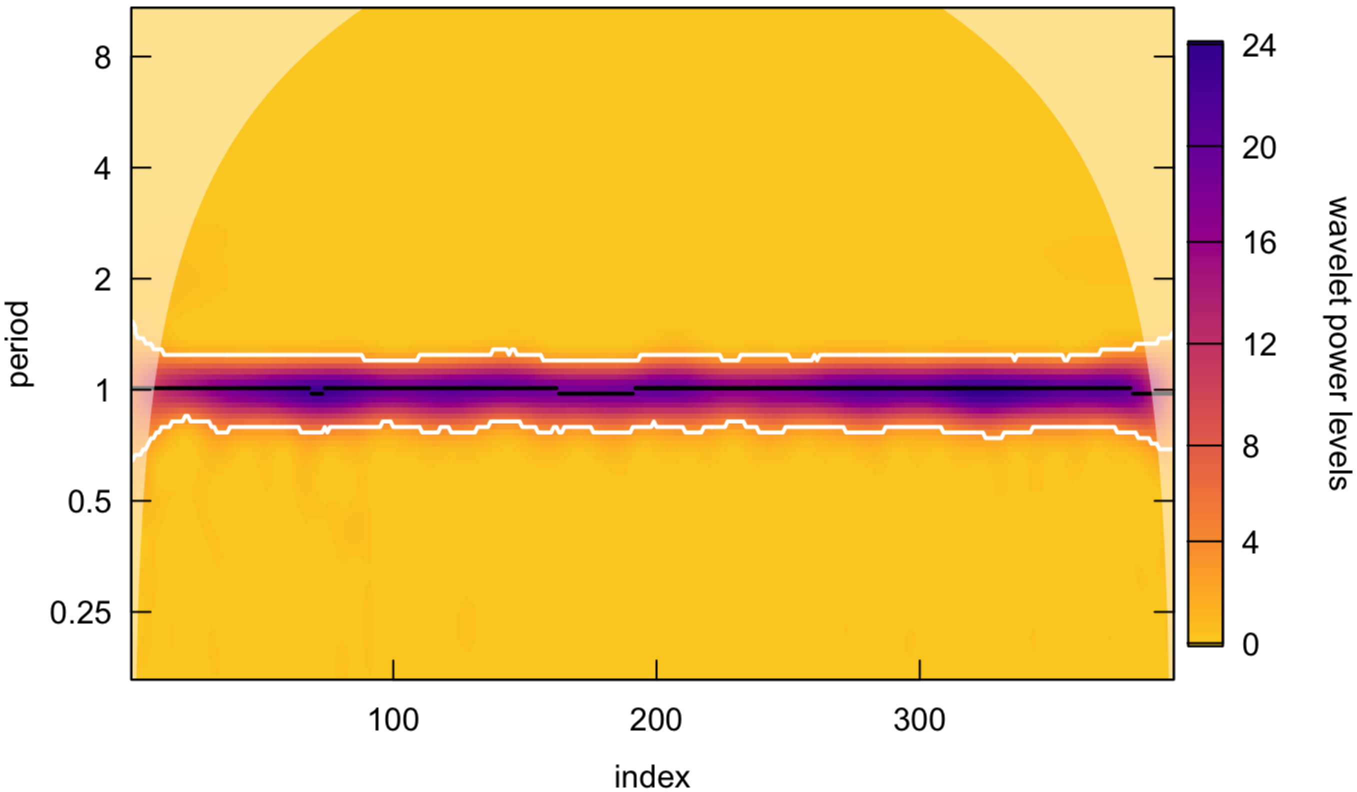
Estimating wavelet transforms in R

Use `analyze.wavelet()` to estimate the wavelet transform

```
w_est <- analyze.wavelet(dat, "tmp",         ## need both df & colname
                         loess.span = 0,     ## no de-trending
                         dt = 1/12,          ## monthly sampling
                         lowerPeriod = 1/6,  ## default = 2*dt
                         n.sim = 100,
                         verbose = FALSE)
```

```
##
  |
  |                                                                      |   0%
  |
  |=                                                                     |   1%
  |
  |=                                                                     |   2%
  |
  |==                                                                    |   3%
  |
  |===                                                                   |   4%
  |
  |====                                                                  |   5%
  |
```

Estimating wavelets in R



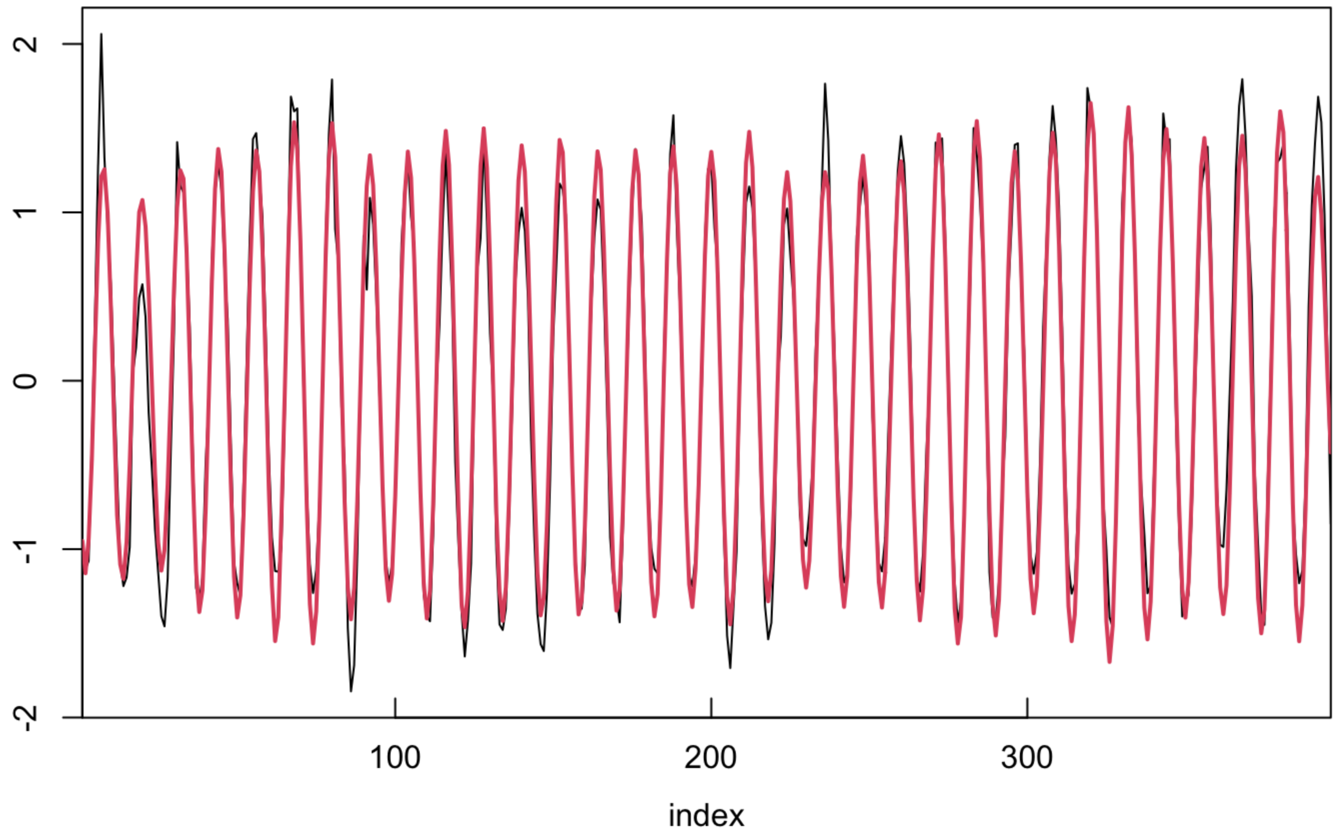
Use `wt.image()` to plot the spectrum

Inverse wavelet transforms

Involves integral calculus

$$f(t) = \frac{1}{C_\psi} \int_a \int_b < f(t), \psi_{a,b}(t) > \psi_{a,b}(t) db \frac{da}{a^2}$$

Inverse wavelet transforms in R



Use `reconstruct()` to get estimate of original time series