STAT302: Time Series Analysis

Chapter 8. Spectrum Analysis

Sangbum Choi, Ph.D

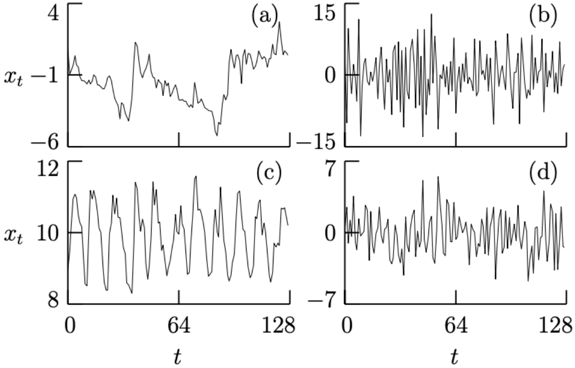
Department of Statistics, Korea University

Outline

Introduction

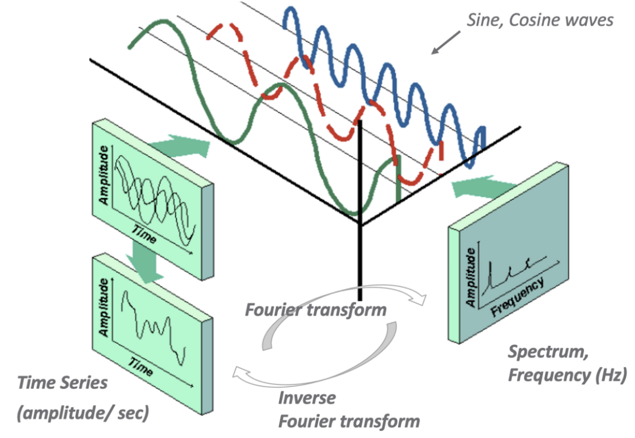
- **Spectrum analysis**, also referred to as **frequency domain analysis** or **spectral density estimation**, is the technical process of decomposing a complex signal into simpler parts.
- The goal of spectral estimation is to estimate the spectral density of a signal from a sequence of time samples of the signal.
- Intuitively speaking, the spectral density characterizes the frequency content of the signal.
- One purpose of estimating the spectral density is to detect any periodicities in the data, by observing peaks at the frequencies corresponding to these periodicities.
- This topic is highly technical and we only cover some introductory materials here.

- Consider four examples of time series $x_1, x_2, ..., x_{127}, x_{128}$.



- Q: How would you describe these 4 series?
- A: Spectral analysis describes x_t's by comparing them to sine and cosine functions.

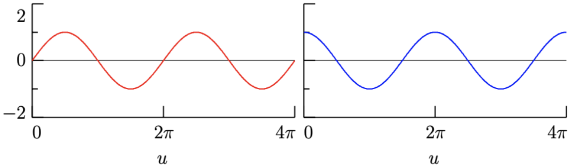
Time domain vs. frequency domain analysis



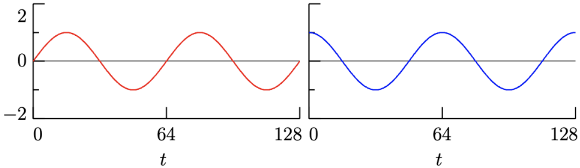
time-series vs. spectrum

Sines and Cosines

- What do sines and cosines have to do with time series?
- Plots of $\sin(u)$ and $\cos(u)$ versus u as u goes from 0 to 4π:

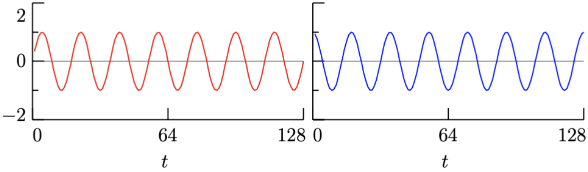


- Let $u = 2\pi \frac{2}{128} t$ for $t = 1, 2, ..., 128$.
- Plots of $\sin(2\pi \frac{2}{128} t)$ and $\cos(2\pi \frac{2}{128} t)$ versus t:



Sines and Cosines

- Now consider $u = 2\pi\frac{7}{128}t$ for $t = 1, 2, .., 128$.
- Plots of $\sin(2\pi\frac{7}{128}t)$ and $\cos(2\pi\frac{7}{128}t)$ versus t:



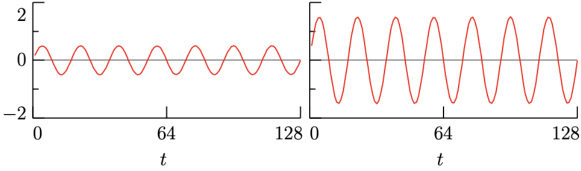
- Now we get series with "7 cycles over time span of 128".
- In general, plots of $x_t = \sin(2\pi\frac{k}{128}t)$ etc versus t have k cycles.
- Here, $\frac{128}{k}$ is called the **period** of x_t as one cycle completes within this time interval.

Sines and Cosines

- The quantity $\frac{k}{128}$ is called **frequency** of sine or cosine and usually use f (or ω) to denote frequency:
  - if k is small, sine time series is said to have *low* frequency
  - if k is large, sine time series is said to have *high* frequency
- **Amplitude** A is the maximum range of variation in

$$x_t = A\sin(2\pi\frac{k}{128}t)$$

- Plots of $0.5\sin(2\pi\frac{7}{128}t)$ and $1.5\sin(2\pi\frac{7}{128}t)$:



Sines and Cosines

- In general, a sine wave of *frequency* ω*, amplitude* A*, and phase* ψ is

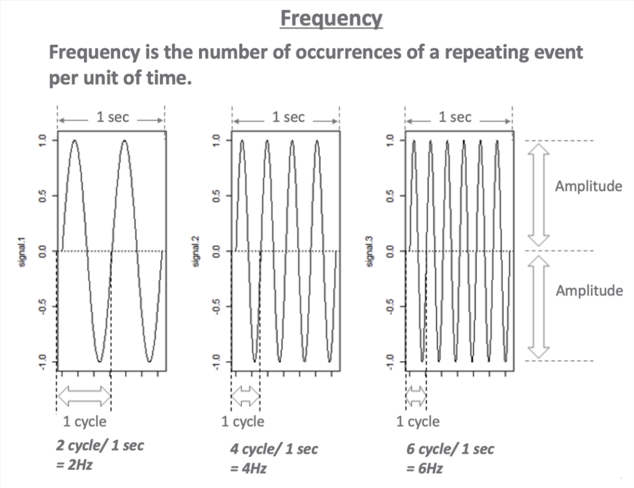
$$y_t = A\sin(\omega t + \psi)$$

- The positive phase shift represents an advance of $\psi/2\pi$ cycles.
- A frequency of f cycles per second is equivalent to ω radians per second, where

$$\omega = 2\pi f \quad \Leftrightarrow \quad f = \frac{\omega}{2\pi}$$

- The mathematics is naturally expressed in radians, but Hz is generally used in physical applications.

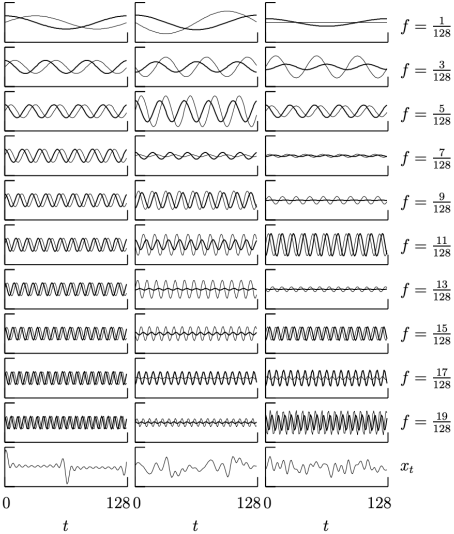
Sines and Cosines



frequency (* https://rfriend.tistory.com)

Spectrum analysis

- Let's add together sines and cosines with different frequencies.
- First column uses sines and cosines of amplitude 1; second and third columns use random amplitudes.



Outline

Spectrum analysis

- **Conclusion:** By summing up lots of sines and cosines with different amplitudes, can get artificial time series that resemble actual time series.
- Given a time series x_t, figure out how to construct it using sines and cosines; i.e., to write

$$x_t = \sum_k a_k \sin\left(2\pi \frac{k}{128} t\right) + b_k \cos\left(2\pi \frac{k}{128} t\right)$$

- The above formula is called **Fourier representation** for a time series, which allows us to reexpress time series in a standard way, i.e., the sum of an infinite number of sine and cosine terms.

Fourier series

- A Fourier series is an approximation to a signal defined for continuous time over a finite period. The signal may have discontinuities.
- Different time series will need different a_k's and b_k's. We can compare different time series by comparing the a_k's and b_k's.
- Q: How do we figure out what a_k's and b_k's should be to form a particular time series?
- A: The answer turns out to be surprisingly simple:

$$a_k = \frac{1}{64}\sum_{t=1}^{128} x_t \sin\left(2\pi \frac{k}{128} t\right) \quad \text{and} \quad b_k = \frac{1}{64}\sum_{t=1}^{128} x_t \cos\left(2\pi \frac{k}{128} t\right)$$

- This calculation is usually performed with the fast fourier transform algorithm (FFT).

Fourier series

- Let $y_1, ..., y_{128}$ and $z_1, ..., z_{128}$ be two time series.
- Let $\bar{y}$ and $\bar{z}$ be their sample means and let σ_y^2 and σ_z^2 be their sample variances.
- Sample correlation coefficient:

$$\hat{\rho} = \frac{\sum_t (y_t - \bar{y})(z_t - \bar{z})}{128\sigma_y\sigma_z} = \frac{1}{128}\frac{\sum_t y_t z_t}{\sigma_y\sigma_z},$$

  where the 2nd equality holds if $\bar{z} = 0$.
- Similarly, a_k is related to strength of linear relationship between $y_t = x_t$ and $z_t = \sin(2\pi \frac{k}{128} t)$, for which $\bar{z}_t = 0$.

Spectrum analysis

- To summarize how important frequency $\frac{k}{128}$ is in

$$x_t = \sum_k a_k \sin\left(2\pi \frac{k}{128} t\right) + b_k \cos\left(2\pi \frac{k}{128} t\right),$$

  let us form $S_k = \frac{1}{2}(a_k^2 + b_k^2)$:
  - if frequency $\frac{k}{128}$ is important, then S_k should be large;
  - if frequency $\frac{k}{128}$ is not important, then S_k should be small.
- S_k over all frequencies $f = \frac{k}{128}$ is called the **spectrum**.

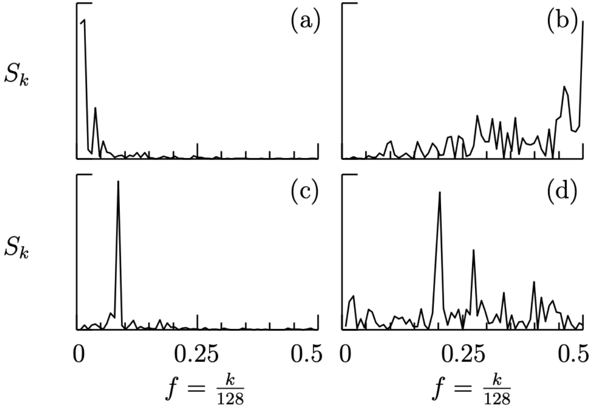
Spectrum analysis

- Fundamental fact about the spectrum:

$$\sum_k S_k = \frac{1}{128} \sum_{t=1}^{128} (x_t - \bar{x})^2 = \sigma_x^2$$

- That is, spectrum breaks sample variance of time series up into pieces, each of which is associated with a particular frequency.
- Spectral analysis is thus an analysis of variance technique, in which σ_x^2 is broken up across different frequencies.

- Here are the spectra for the four examples:

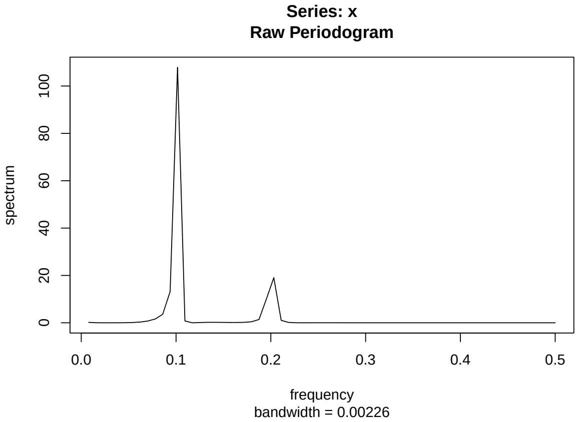


Outline

Sines and Cosines

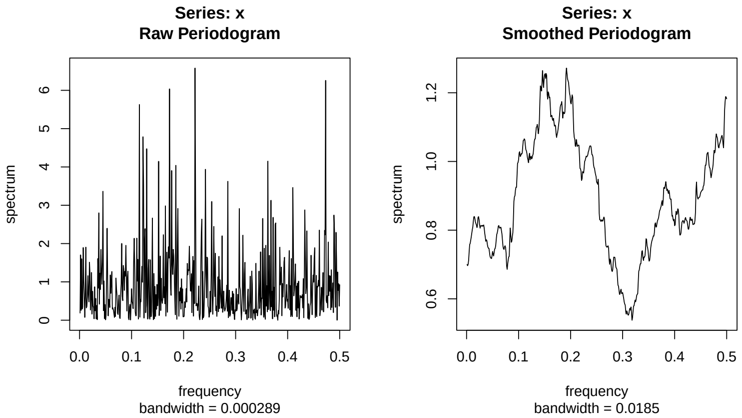
Two frequency peaks are expected at 0.1 and 0.2.

```
t = 1:128
y = 2*sin(2*pi*0.1*t) + cos(2*pi*0.2*t)
spectrum(y, log=c("no"))
```



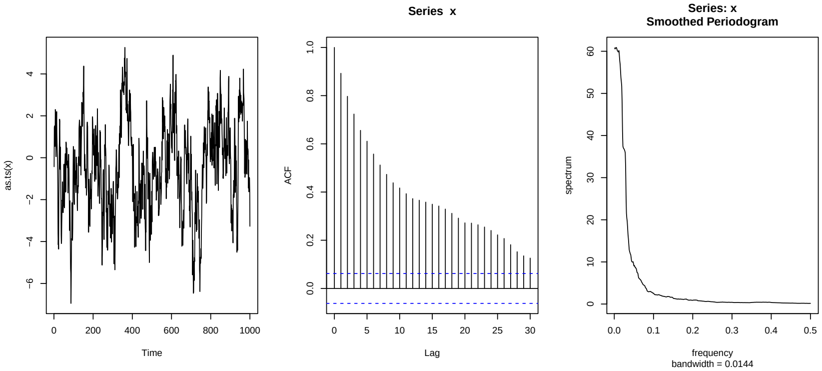
White noises

```r
par(mfrow=c(1,2))
x = rnorm(1000)
spectrum(x, log = c("no"))
spectrum(x, span = 65, log = c("no"))
```



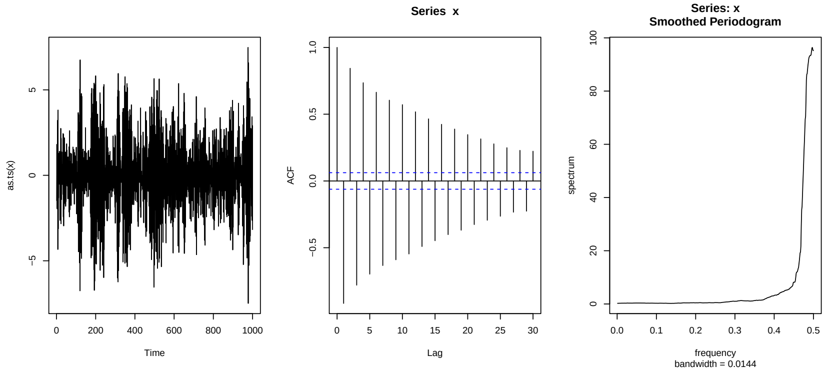
AR(1): Positive coefficient

```
par(mfrow=c(1,3))
x = arima.sim(n=1000, list(ar=0.9))
plot(as.ts(x))
acf(x)
spectrum(x, span = 50, log = c("no"))
```



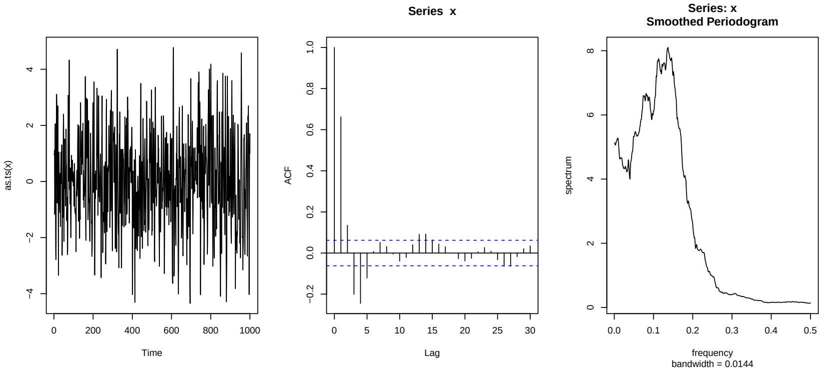
AR(1): Negative coefficient

```r
par(mfrow=c(1,3))
x = arima.sim(n=1000, list(ar=-0.9))
plot(as.ts(x))
acf(x)
spectrum(x, span = 50, log = c("no"))
```



AR(2)

```r
par(mfrow=c(1,3))
x = arima.sim(n=1000, list(ar=c(1,-0.5)))
plot(as.ts(x))
acf(x)
spectrum(x, span = 50, log = c("no"))
```



Sunspot data

```
library(fpp2)
par(mfrow=c(1,3))
plot(sunspotarea); acf(sunspotarea)
spectrum(sunspotarea, span = 50, log = c("no"))
```