

# Modeling and Analysis of Time Series Data

## Chapter 7: Introduction to time series analysis in the frequency domain

Edward L. Ionides

# Outline

- 1 Frequency components
  - Eigenvalues and eigenvectors of a stationary covariance matrix
- 2 The Fourier transform
  - A normal approximation
- 3 The periodogram to estimate the spectral density
- 4 Frequency domain data analysis
  - Smoothing the periodogram
  - Tapering before calculating the periodogram
  - Fitting an AR model to estimate the spectrum

# Frequency components of a time series

- ① A time series dataset (like any other sequence of numbers) can be written as a sum of sine and cosine functions with varying frequencies.
- ② This is called the **Fourier representation** or **Fourier transform** of the data.
- ③ The coefficients corresponding to the sine and cosine at each frequency are called **frequency components** of the data.
- ④ Looking at which frequencies have large and small components can help to identify appropriate models.
- ⑤ Looking at the frequency components present in our models can help to assess whether they are doing a good job of describing our data.

# What is the spectrum of a time series model?

- We begin by reviewing eigenvectors and eigenvalues of covariance matrices. This eigen decomposition also arises elsewhere in statistics, e.g. principle component analysis.
- A univariate time series model is a vector-valued random variable  $Y_{1:N}$  which we suppose has a covariance matrix  $V$  which is an  $N \times N$  matrix with entries  $V_{mn} = \text{Cov}(Y_m, Y_n)$ .
- $V$  is a non-negative definite symmetric matrix, and therefore has  $N$  non-negative eigenvalues  $\lambda_1, \dots, \lambda_N$  with corresponding eigenvectors  $u_1, \dots, u_N$  such that

$$Vu_n = \lambda_n u_n. \quad (1)$$

- A basic property of these eigenvectors is that they are orthogonal, i.e.,

$$u_m^T u_n = 0 \text{ if } m \neq n. \quad (2)$$

- We may work with **normalized** eigenvectors that are scaled such that  $u_n^T u_n = 1$ .

- We can also check that the components of  $Y$  in the directions of different eigenvectors are uncorrelated.
- Since  $\text{Cov}(AY, BY) = A\text{Cov}(Y, Y)B^T$ , we have

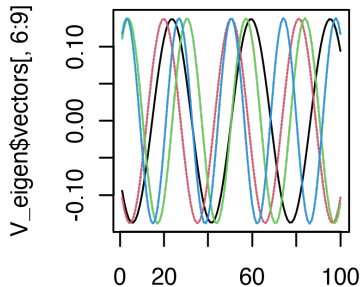
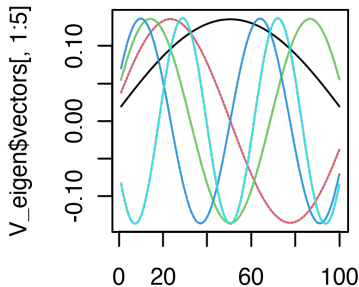
$$\begin{aligned}\text{Cov}(u_m^T Y, u_n^T Y) &= u_m^T \text{Cov}(Y, Y) u_n \\ &= u_m^T V u_n \\ &= \lambda_n u_m^T u_n \\ &= \begin{cases} \lambda_n & \text{if } m = n \\ 0 & \text{if } m \neq n \end{cases}\end{aligned}$$

For the last equality, we have supposed that the eigenvectors are normalized.

- If we knew  $V$ , we could convert the model to a representation where the observable random variables are uncorrelated.
- Transforming the data into its components in the directions of the eigenvectors of the model allows us to use an uncorrelated model. In the Gaussian case, we have independence.

# Eigenvectors for the covariance matrix of an AR(1) model with $N = 100$ and $\phi = 0.8$

```
N <- 100; phi <- 0.8; sigma <- 1
V <- matrix(NA,N,N)
for(m in 1:N) for(n in 1:N) V[m,n]<-sigma^2*phi^abs(m-n)/(1-phi^2)
V_eigen <- eigen(V,symmetric=TRUE)
matplot(V_eigen$vectors[,1:5],type="l")
matplot(V_eigen$vectors[,6:9],type="l")
```



## Eigenvalues for the covariance matrix of an AR(1) model with $N = 100$ and $\phi = 0.8$

- We see that the eigenvectors, plotted as functions of time, look like sine wave oscillations.
- The eigenvalues are

```
round(V_eigen$values[1:9],2)
[1] 24.59 23.44 21.73 19.70 17.57 15.51 13.61 11.91 10.42
```

- We see that the eigenvalues are decreasing. For this model, the components of  $Y_{1:N}$  with highest variance correspond to long-period oscillations.
- Are the sinusoidal eigenvectors a special feature of this particular time series model, or something more general?

# The eigenvectors for a long stationary time series model

- Suppose  $\{Y_n, -\infty < n < \infty\}$  has a stationary autocovariance function  $\gamma_h$ .
- We write  $\Gamma$  for the infinite matrix with entries

$$\Gamma_{m,n} = \gamma_{m-n} \quad \text{for all integers } m \text{ and } n. \quad (3)$$

- An infinite eigenvector is a sequence  $u = \{u_n, -\infty < n < \infty\}$  with corresponding eigenvalue  $\lambda$  such that

$$\Gamma u = \lambda u, \quad (4)$$

or, writing out the matrix multiplication explicitly,

$$\sum_{n=-\infty}^{\infty} \Gamma_{m,n} u_n = \lambda u_m \quad \text{for all } m. \quad (5)$$

- We look for a sinusoidal solution,  $u_n = e^{2\pi i \omega n}$ , where  $\omega$  is cycles per unit time.



$$\begin{aligned}\sum_{n=-\infty}^{\infty} \Gamma_{m,n} u_n &= \sum_{n=-\infty}^{\infty} \gamma_{m-n} u_n \\ &= \sum_{h=-\infty}^{\infty} \gamma_h u_{m-h} \quad \text{setting } h = m - n \\ &= \sum_{h=-\infty}^{\infty} \gamma_h e^{2\pi i \omega (m-h)} \\ &= e^{2\pi i \omega m} \sum_{h=-\infty}^{\infty} \gamma_h e^{-2\pi i \omega h} \\ &= u_m \lambda(\omega) \quad \text{for } \lambda(\omega) = \sum_{h=-\infty}^{\infty} \gamma_h e^{-2\pi i \omega h}\end{aligned}$$

**Question 7.1.** Why does this calculation show that  $u_n(\omega) = e^{2\pi i \omega n}$  is an eigenvector for  $\Gamma$  for any choice of  $\omega$ .

- The eigenvalue at frequency  $\omega$  is

$$\lambda(\omega) = \sum_{h=-\infty}^{\infty} \gamma_h e^{-2\pi i \omega h} \quad (6)$$

- Viewed as a function of  $\omega$ , this is called the **spectral density function**.
- $\lambda(\omega)$  is the **Fourier transform** of  $\gamma_h$ .
- An integral version of (6) is used in applied math and engineering:

$$\lambda(\omega) = \int_{-\infty}^{\infty} \gamma(x) e^{-2\pi i \omega x} dx. \quad (7)$$

- We obtain (6) from (7) when  $\gamma(h)$  has a point mass  $\gamma_h$  when  $h$  is an integer, and  $\gamma(x) = 0$  for non-integer  $x$ .

- It was convenient to do this calculation with complex exponentials. However, writing

$$e^{2\pi i \omega n} = \cos(2\pi \omega n) + i \sin(2\pi \omega n), \quad (8)$$

and noting that  $\gamma_h$  is real, we see that the real and imaginary parts of  $\lambda(\omega) = \sum_{h=-\infty}^{\infty} \gamma_h e^{-2\pi i \omega h}$  give us two real eigenvectors,  $\cos(2\pi \omega n)$  and  $\sin(2\pi \omega n)$ .

**Question 7.2.** Review: how would you demonstrate the correctness of the identity  $e^{2\pi i \omega} = \cos(2\pi \omega) + i \sin(2\pi \omega)$ .

- Assuming that this computation for an infinite sum represents a limit of increasing dimension for finite matrices, we have found that the eigenvectors for any long, stationary time series model are approximately sinusoidal.
- For the finite time series situation, we only expect  $N$  eigenvectors for a time series of length  $N$ . We have one eigenvector for  $\omega = 0$ , two eigenvectors corresponding to sine and cosine functions with frequency

$$\omega_n = n/N, \text{ for } 0 < n < N/2, \quad (9)$$

and, if  $N$  is even, a final eigenvector with frequency

$$\omega_{(N/2)} = 1/2. \quad (10)$$

- These sine and cosine vectors are the **Fourier basis**.
- The time series  $y_{1:N}^*$  is the **time domain** representation of the data. Transforming to the Fourier basis gives the **frequency domain** representation.

# Frequency components and the Fourier transform

- The **frequency components** of  $Y_{1:N}$  are the components in the directions of these eigenvectors, given by

$$C_n = \frac{1}{\sqrt{N}} \sum_{k=1}^N Y_k \cos(2\pi\omega_n k) \text{ for } 0 \leq n \leq N/2,$$

$$S_n = \frac{1}{\sqrt{N}} \sum_{k=1}^N Y_k \sin(2\pi\omega_n k) \text{ for } 1 \leq n \leq N/2.$$

- Similarly, the **frequency components** of data  $y_{1:N}^*$  are

$$c_n = \frac{1}{\sqrt{N}} \sum_{k=1}^N y_k^* \cos(2\pi\omega_n k) \text{ for } 0 \leq n \leq N/2,$$

$$s_n = \frac{1}{\sqrt{N}} \sum_{k=1}^N y_k^* \sin(2\pi\omega_n k) \text{ for } 1 \leq n \leq N/2.$$

- The frequency components of the data can be written as real and imaginary parts of the **discrete Fourier transform**,

$$\begin{aligned} d_n &= \frac{1}{\sqrt{N}} \sum_{k=1}^N y_k^* e^{-2\pi i n k / N} \\ &= c_n - i s_n \end{aligned}$$

- The normalizing constant of  $1/\sqrt{N}$  is convenient for a central limit theorem.
- Various choices about signs and factors of  $2\pi$ ,  $\sqrt{2\pi}$  and  $\sqrt{N}$  can be made in the definition of the Fourier transform. For example, the `fft` command in R does not include this constant.
- `fft` is an implementation of the fast Fourier transform algorithm, which enables computation of all the frequency components with order  $N \log(N)$  computation. Computing the frequency components may appear to require a matrix multiplication involving order  $N^3$  additions and multiplications. When  $N = 10^5$  or  $N = 10^6$  this difference becomes important!

- The first frequency component,  $C_0$ , is a special case, since it has mean  $\mu = \mathbb{E}[Y_n]$  whereas the other components have mean zero.
- In practice, we subtract a mean before computing the frequency components, which is equivalent to removing the frequency component for frequency zero.
- The frequency components  $(C_{0:N/2}, S_{1:N/2})$  are asymptotically uncorrelated. They are constructed as a sum of a large number of terms, with the usual  $1/\sqrt{N}$  scaling for a central limit theorem. So, it may not be surprising that a central limit theorem applies, giving asymptotic justification for the following normal approximation.
- Moving to the frequency domain (i.e., transforming the data to its frequency components) has **decorrelated** the data. Statistical techniques based on assumptions of independence are appropriate when applied to frequency components.

# Normal approximation for the frequency components

- $(C_{1:N/2}, S_{1:N/2})$  are approximately independent, mean zero, Normal random variables with

$$\text{Var}(C_n) = \text{Var}(S_n) \approx 1/2\lambda(\omega_n). \quad (11)$$

- $C_0/\sqrt{N}$  is approximately Normal, mean  $\mu$ , independent of  $(C_{1:N/2}, S_{1:N/2})$ , with

$$\text{Var}(C_0/\sqrt{N}) \approx \lambda(0)/N. \quad (12)$$

- It follows from the normal approximation that, for  $1 \leq n \leq N/2$ ,

$$C_n^2 + S_n^2 \approx \lambda(\omega_n) \frac{\chi_2^2}{2}, \quad (13)$$

where  $\chi_2^2$  is a chi-squared random variable on two degrees of freedom.

- Taking logs, we have

$$\log(C_n^2 + S_n^2) \approx \log \lambda(\omega_n) + \log(\chi_2^2/2). \quad (14)$$



# The periodogram

- The chi-squared property in (13) motivates the **periodogram**,

$$I_n = c_n^2 + s_n^2 = |d_n|^2 \quad (15)$$

as an estimator of the spectral density.

- From (14),  $\log I_n$  is as an estimator of the log spectral density with a convenient statistical property: asymptotically independent, identically distributed errors at each Fourier frequency.
- Therefore, a signal-plus-white-noise model is appropriate for estimating the log spectral density using the log periodogram.
- The periodogram is an **inconsistent estimator** of the spectrum. We can smooth the periodogram to borrow strength between nearby frequencies.

# Interpreting the spectral density as a power spectrum

- The power of a wave is proportional to the square of its amplitude.
- The spectral density gives the mean square amplitude of the components at each frequency, and therefore gives the expected power.
- The spectral density function can therefore be called the **power spectrum**.

**Question 7.3.** Consider the AR(1) model,  $\phi(B)Y_n = \epsilon_n$  with  $\phi(B) = 1 - \phi_1 B$  and  $\epsilon_n \sim \text{WN}(\sigma^2)$ , i.e., white noise with variance  $\sigma^2$ . Show that the spectrum of  $Y$  is

$$\lambda(\omega) = \frac{\sigma^2}{|\phi(e^{2\pi i\omega})|^2} = \frac{\sigma^2}{1 + \phi_1^2 - 2\phi_1 \cos(2\pi\omega)}. \quad (16)$$

# ARMA models have a rational spectrum

- The calculation for the AR(1) model generalizes. We give the result without proof.
- Let  $Y_n$  be an ARMA(p,q) model,  $\phi(B)Y_n = \psi(B)\epsilon_n$  with  $\epsilon_n \sim \text{WN}(\sigma^2)$ . The spectrum of  $Y$  is

$$\lambda(\omega) = \sigma^2 \left| \frac{\psi(e^{2\pi i\omega})}{\phi(e^{2\pi i\omega})} \right|^2. \quad (17)$$

- The so-called **rational spectrum** of ARMA models is computationally convenient.
- A stationary, causal ARMA model cannot have roots on the unit circle. If a root approaches the unit circle, the denominator in (17) becomes close to zero.
- The special case of  $\phi(x) = \psi(x) = 1$  gives  $\lambda(\omega) = \sigma^2$ . **White noise has a constant spectrum**, matching the analogy that white light has uniform intensity across the visible light spectrum.

# Michigan winters revisited: Frequency domain methods

```
y <- read.table(file="ann_arbor_weather.csv",header=TRUE)
head(y[,1:9],3)
```

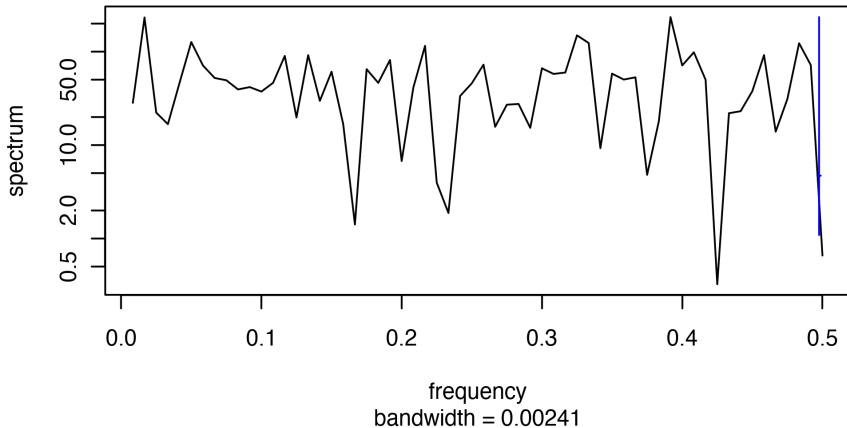
Year	Low	High	Hi_min	Lo_max	Avg_min	Avg_max	Mean	Precip
1900	-7	50	36	12	18	34.7	26.3	1.06
1901	-7	48	37	20	17	31.8	24.4	1.45
1902	-4	41	27	11	15	30.4	22.7	0.60

- We have to deal with the NA measurement for 1955. A simple approach is to replace the NA by the mean.
- What other approaches can you think of for dealing with this missing observation?
- What are the strengths and weaknesses of these approaches?

```
low <- y$Low
low[is.na(low)] <- mean(low, na.rm=TRUE)
```

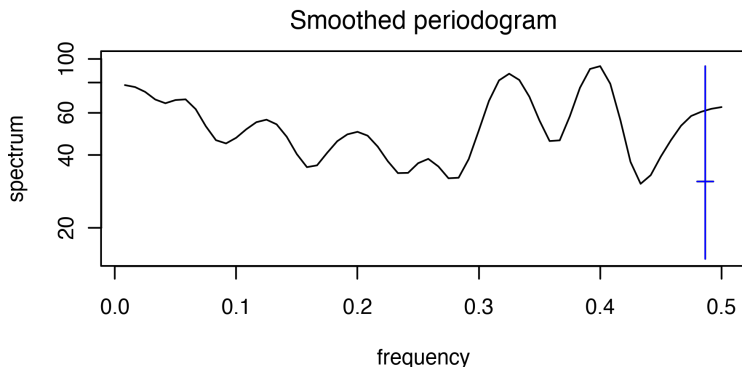
```
spectrum(low, main="Unsmoothed periodogram")
```

Unsmoothed periodogram



- To smooth, we use the default periodogram smoother in R

```
spectrum(low, spans=c(3,5,3), main="Smoothed periodogram",  
ylim=c(15,100))
```



- The bar is a 95% pointwise confidence interval which we can slide to any frequency of interest.
- The chi-squared property (14) means this CI is the same width for each frequency, on the log scale. Note it is asymmetric.

**Question 7.4.** What is the default periodogram smoother in R?

**Question 7.5.** How should we use it?

**Question 7.6.** Why is that default chosen?



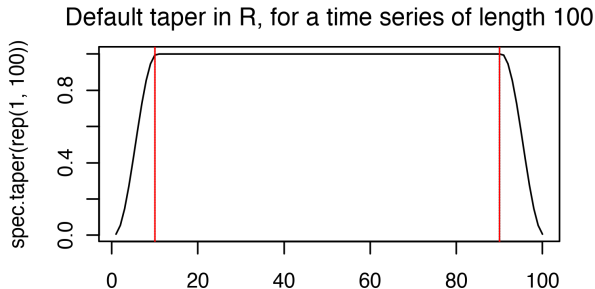
## More details on computing and smoothing the periodogram

- To see what R actually does to compute and smooth the periodogram, type `?spectrum`.
- This will lead you to type `?spec.pgram`.
- You will see that, by default, R removes a linear trend, fitted by least squares. This may often be a sensible thing to do. Why?
- You will see that R then multiplies the data by a quantity called a **taper**, computed by `spec.taper`.
- The taper smooths the ends of the time series and removes high-frequency artifacts arising from an abrupt start and end to the time series.
- Formally, from the perspective of the Fourier transform, the time series takes the value zero outside the observed time points  $1:N$ . The sudden jump to and from zero at the start and end produces unwanted effects in the frequency domain.

The default taper in R smooths the first and last  $p = 0.1$  fraction of the time points, by modifying the detrended data  $y_{1:N}^*$  to tapered version  $z_{1:N}$  defined by

$$z_n = \begin{cases} y_n^* (1 - \cos(\pi n / Np)) / 2 & \text{if } 1 \leq n < Np \\ y_n^* & \text{if } Np \leq n \leq N(1 - p) \\ y_n^* (1 - \cos(\pi [N + 1 - n] / Np)) / 2 & \text{if } N(1 - p) < n \leq N \end{cases}$$

```
plot(spec.taper(rep(1,100)),type="l",
     main="Default taper in R, for a time series of length 100")
abline(v=c(10,90),lty="dotted",col="red")
```

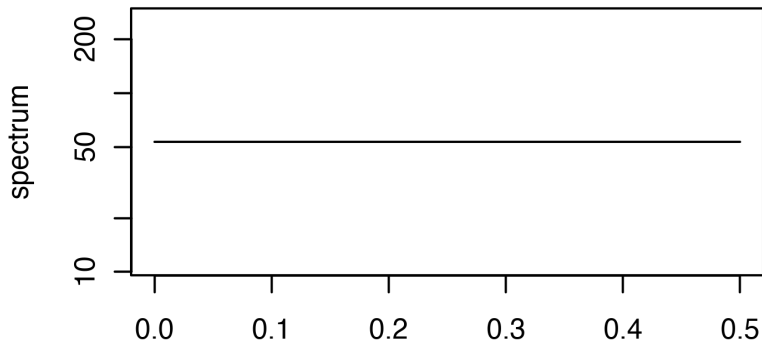


## Spectral density estimation by fitting a model

Another standard way to estimate the spectrum is to fit an AR( $p$ ) model with  $p$  selected by AIC.

```
spectrum(low,method="ar",  
  main="Spectrum estimated via AR model picked by AIC")
```

Spectrum estimated via AR model picked by AIC



## Units of frequency and period

- When we call  $\omega$  the frequency in cycles per unit time, we really mean **cycles per unit observation**.
- Suppose the time series consists of equally spaced observations, with  $t_n - t_{n-1} = \Delta$  years. Then, the frequency is  $\omega/\Delta$  **cycles per year**.
- The **period** of an oscillation is the time for one cycle,

$$\text{period} = \frac{1}{\text{frequency}}. \quad (18)$$

- When the observation intervals have a time unit (years, seconds, etc) we usually use that unit for the period, and its inverse for the frequency.

## Further reading

- Sections 4.1 to 4.3 of Shumway and Stoffer (2017) cover similar topics to this chapter.

# References and Acknowledgements

Shumway RH, Stoffer DS (2017). *Time Series Analysis and its Applications: With R Examples*. Springer.

- Compiled on February 1, 2022 using R version 4.1.2.
- Licensed under the [Creative Commons Attribution-NonCommercial](#) license. Please share and remix non-commercially, mentioning its origin.
- We acknowledge [previous versions of this course](#).

