Objectives

This course emphasizes time domain analysis of time series, but we also want to be able to present and interpret the frequency domain properties of our time series models and data.

1. Looking at the frequency components present in our data can help to identify appropriate models.

2. 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 $\lambda_1, \ldots, \lambda_N$ with corresponding eigenvectors $u_1, \ldots, u_N$ such that
$$V u_n = \lambda_n u_n.$$

- A basic property of these eigenvectors is that they are orthogonal, i.e.,
$$u_m^{\mathrm{T}} u_n = 0 \text{ if } m \neq n.$$

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

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

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

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

- Thus, if we knew V, we could convert the model to a representation where the observable random variables are uncorrelated.
- Specifically, we could transform the data into its components in the directions of the eigenvectors of the model. An uncorrelated (or, in the Gaussian case, independent) model would then become appropriate for this transformation of the data.

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

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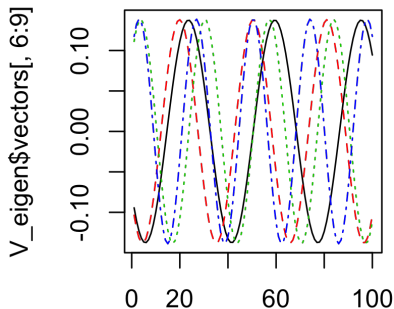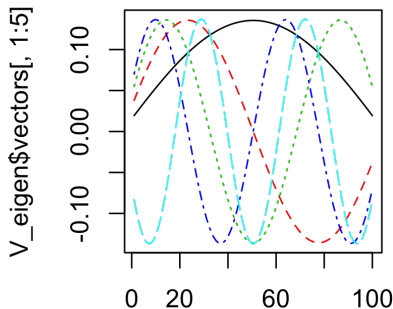
Eigenvalues for the covariance matrix of an AR(1) model with $N = 100$ and $\phi = 0.8$

- 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 γ_h.

- We write Γ for the infinite matrix with entries

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

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

$$\Gamma u = \lambda u,$$

or, writing out the matrix multiplication explicitly,

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

- Now, let's look for a sinusoidal solution, $u_n = e^{i\omega n}$.

$$\sum_{n=-\infty}^{\infty} \Gamma_{m,n} u_n = \sum_{n=-\infty}^{\infty} \gamma_{m-n} u_n$$

$$= \sum_{h=-\infty}^{\infty} \gamma_h u_{m-h} \quad \text{setting } h = m - n$$

$$= \sum_{h=-\infty}^{\infty} \gamma_h e^{i\omega(m-h)}$$

$$= e^{i\omega m} \sum_{h=-\infty}^{\infty} \gamma_h e^{-i\omega h}$$

$$= u_m \lambda(\omega) \text{ for } \lambda = \sum_{h=-\infty}^{\infty} \gamma_h e^{-i\omega h} = \lambda(\omega)$$

Question 7.1. Why does this calculation show that $u_n(\omega) = e^{i\omega n}$ is an eigenvector for Γ for any choice of ω. *This argument holds for any ω, and meets the definition of an eigenvector, ignoring issues related to infinite sums.*

- The corresponding eigenvalue function,

$$\lambda(\omega) = \sum_{h=-\infty}^{\infty} \gamma_h e^{-i\omega h},$$

this is the discrete FT, corresponding to the integral definition.

is called the **spectral density function**. It is calculated as the **Fourier transform** of γ_h at frequency ω.

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

$$e^{i\omega n} = \cos(\omega n) + i\sin(\omega n)$$

we see that the real and imaginary parts of this calculation in fact give us two real eigenvectors, $\cos(\omega n)$ and $\sin(\omega n)$.

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

Construct Taylor series expansions of each term.

this is just on eigenvector, when we think of the value u_n *as a function of* n.

- Assuming that this computation for an infinite sum represents a limit of increasing dimension for finite matrices, we have found that the eigenfunctions 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

$$\omega_n = 2\pi n/N, \text{ for } 0 < n < N/2,$$

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

$$\omega_{(N/2)} = \pi.$$

- These sine and cosine vectors are called the **Fourier basis**.

Frequency components of the data and their representation via the Fourier transform

- The **frequency components** of $Y_{1:N}$ are the components in the directions of these eigenvectors. Equivalently, we could say they are the representation of $Y_{1:N}$ in the Fourier basis. Specifically, we write

$$
\begin{aligned}
C_n &= \frac{1}{\sqrt{N}} \sum_{k=1}^{N} Y_k \cos(\omega_n k) \text{ for } 0 \le n \le N/2, \\
S_n &= \frac{1}{\sqrt{N}} \sum_{k=1}^{N} Y_k \sin(\omega_n k) \text{ for } 1 \le n \le N/2.
\end{aligned}
$$

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

$$
\begin{aligned}
c_n &= \frac{1}{\sqrt{N}} \sum_{k=1}^{N} y_k \cos(\omega_n k) \text{ for } 0 \le n \le N/2, \\
s_n &= \frac{1}{\sqrt{N}} \sum_{k=1}^{N} y_k \sin(\omega_n k) \text{ for } 1 \le n \le N/2.
\end{aligned}
$$

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

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

- Here, we have introduced a normalizing constant of $1/\sqrt{N}$. There are various choices about signs and factors of 2π, $\sqrt{2\pi}$ and $\sqrt{N}$ that can—and are—made in the definition of the Fourier transform in various situations.
- One should try to be consistent, and also be careful: the `fft` command in R, for example, 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. At first consideration, computing the frequency components appears to require a matrix multiplication involving order N^2 additions and multiplications. When $N = 10^5$ or $N = 10^6$ this difference becomes important!

- The first frequency component, C_0, is something of 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 periodogram, 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

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

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

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

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

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

where χ_2^2 is a chi-squared random variable on two degrees of freedom.

- Taking logs, we have

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

Handwritten annotations:

note: $|a+ib| = \sqrt{a^2 + b^2}$

$\text{Var}\left(C_n + iS_n\right)$
$= \text{Var}(C_n) + \text{Var}(S_n)$

noting that variance for a complex R.V. is defined as

$\text{Var}(X) = E\left\{(X-\bar{X})(X-\bar{X})^{*}\right\}$

where $*$ is the complex conjugate transpose.

- These results motivate consideration of the **periodogram**,

$$I_n = c_n^2 + s_n^2 = \left| d_n \right|^2$$

  as an estimator of the spectral density.
- $\log I_n$ can be modeled as an estimator of the log spectral density with independent, identically distributed errors.
- We see from the normal approximation that a signal-plus-white-noise model is appropriate for estimating the log spectral density using the log periodogram.

- 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**.

method (i): take the Fourier transform of the ACF.

Question 7.3. compute the spectrum of an AR(1) model.

Method (ii) using convolution. $(1 - \phi B) X_n = \varepsilon_n$

this can be written as a convolution

$$X_n * h_n = \varepsilon_n \quad \text{①}$$

where $h_0 = 1, h_1 = \phi, h_k = 0$ for $k \neq \{0, 1\}$ and

$$X_n * h_n = \sum_{k=0}^{\infty} X_{n-k} h_k. \quad \text{A property of convolutions}$$

is $FT(X_n * h_n) = FT(X_n) . FT(h_n)$.

Take the FT of ①:

$$FT(X_n) . FT(h_n) = FT(\varepsilon_n) \quad \text{②}$$

Question 7.4. compute the spectrum of the MA(q) moving mean,

$$Y_n = \frac{1}{q+1} \sum_{k=0}^{q} \epsilon_{n-k}.$$

Question 7.3 continued.

$$FT(h_n) = \sum h_n e^{i\omega n} = 1 - \phi e^{i\omega}$$

Now use the periodogram property for both sides of ②.
The power spectrum for white noise (var(ϵ_n)=σ^2) is

$$\lambda_{\epsilon}(\omega) = \sigma^2 \quad [\text{CHECK}] \; \text{*}$$

So,
$$|FT(X_n)|^2 \cdot |1 - \phi e^{i\omega}|^2 = |FT(\epsilon_n)|^2$$

which implies
$$\lambda_x(\omega) = \frac{\sigma^2}{(1 - \phi e^{i\omega})(1 - \phi e^{-i\omega})} = \frac{\sigma^2}{|\Phi(e^{i\omega})|^2}$$

$$\text{where } \Phi(x) = 1 - \phi x$$

Some data analysis using the frequency domain: Michigan winters revisited

```
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
## 1 1900  -7   50     36     12      18    34.7 26.3   1.06
## 2 1901  -7   48     37     20      17    31.8 24.4   1.45
## 3 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")
```

note: the unsmoothed periodogram is an inconsistent estimator of the spectral density — the error term does not vanish as n gets large.

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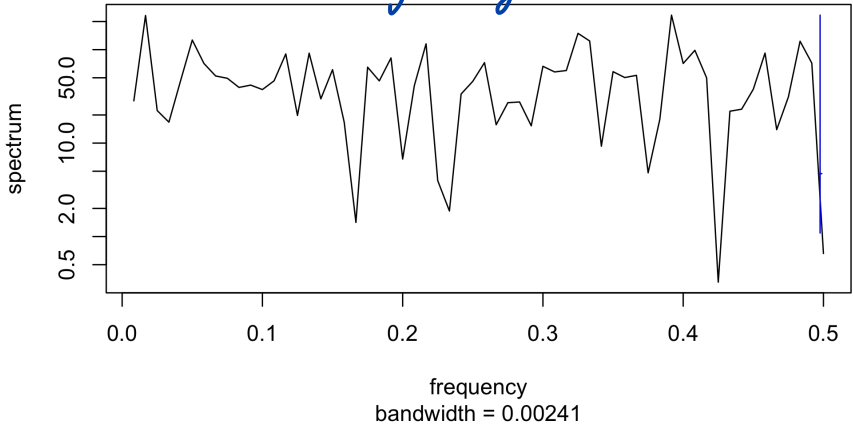
- To smooth, we use the default periodogram smoother in R

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

smoothing parameters. This smooths 3 times with different moving average smoothers.

Smoothed periodogram

The periodogram is consistent with white noise... all points within a 95% CI of a constant spectrum.
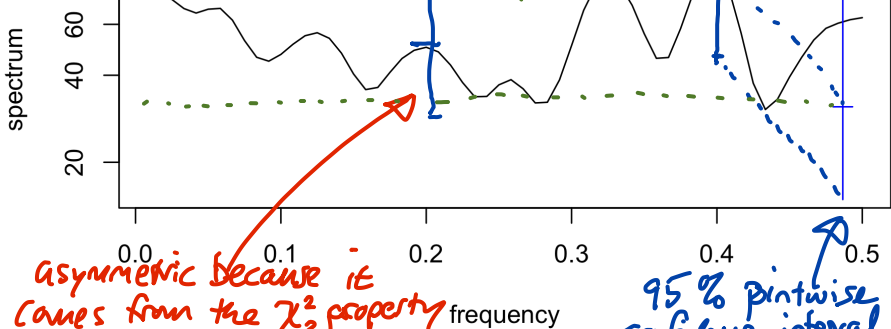
asymmetric because it comes from the χ^2_2 property of each term in the log periodogram.

95% pointwise confidence interval which we can slide to any frequency of interest.

Question 7.5. What is the default periodogram smoother in R?

A repeated moving average for each "span".

Question 7.6. How should we use it?

"It is often good to use 2 or more spans, usually odd"

Question 7.7. Why is that default chosen?

tradition

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

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

```
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")
```


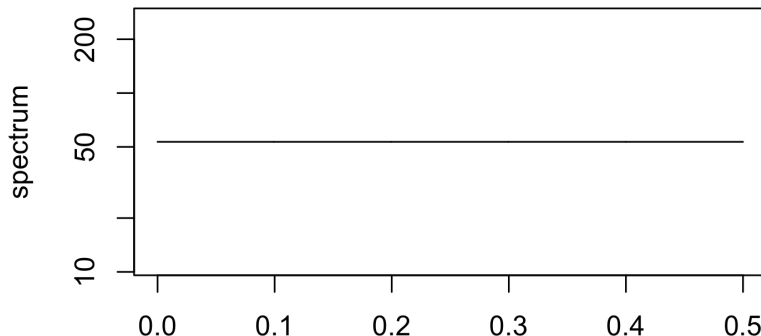Default taper in R, for a time series of length 100

sometimes called a window function.
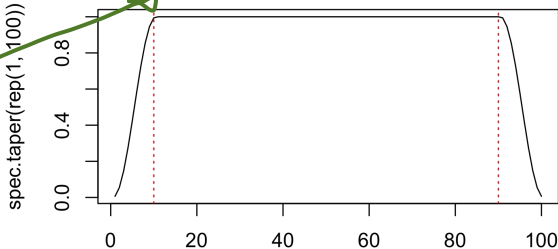
Spectral density estimation by fitting a model

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

- It is always good practice to be explicit about the units of quantities. For frequency domain analysis there are different options for units of frequency.
- For a frequency component corresponding to $\sin(2\pi\omega n)$ or $\cos(2\pi\omega n)$, we say that the frequency is ω **cycles per unit time**.
- Suppose the time series consists of equally spaced observations, with $t_n - t_{n-1} = \Delta$ years. Then we say that the frequency is ω/Δ **cycles per year**.
- For a frequency component corresponding to $\sin(\nu t)$ or $\cos(\nu t)$, we say that the frequency is ν **radians per unit time**.

- The **period** of an oscillation is the time for one cycle. So, when frequency is measured in cycles per time, we have

$$\text{period} = \frac{1}{\text{frequency}}.$$

Thus, for a frequency component corresponding to $\sin(2\pi\omega n)$ or $\cos(2\pi\omega n)$, the period is $1/\omega$ observation intervals.

- When the observation intervals have constant time length (years, seconds, etc) we usually use those units for the period.

Acknowledgments and License

- These notes build on previous versions at `ionides.github.io/531w16` and `ionides.github.io/531w18`.

- Licensed under the Creative Commons attribution-noncommercial license, `http://creativecommons.org/licenses/by-nc/3.0/`. Please share and remix noncommercially, mentioning its origin.