Notes for Tuesday, April 14, 1998

Cyclic Behavior of Time Series: the Spectrum and the Periodogram

Many times series show cyclic fluctuations (hours, days, weeks, months, seasons, years), but the underlying sine and cosine functions to use to model this are usually not obvious.

One way to approach this problem is to look at the *discrete Fourier transform (DFT)* of the time series. The sequence of squared Fourier coefficients in the DFT is called the *spectrum* of the time series. The *periodogram* is an estimate of the spectrum, based on the DFT of the observed time series, that can help us to see what the important cycles in a time series are.

This sort of analysis is called "spectral analysis", or "looking at the time series in the frequency domain." We will first look at the spectrum and periodogram and spectrum from the point of view of Fourier analysis, and then we will look at an interesting connection with linear regression. Finally we will look at some examples.

Short Digression: Fourier Analysis of $f(t)$, $t \in [0, 1]$

A standard theorem from differential equations, advanced calculus, or even a sophomore/junior linear algebra class, says that any reasonable real function $f(t)$ with domain $[0, 1]$ (or periodic function with period 1) has a Fourier series representation,

$$f(t) = \sum_{k=1}^{\infty} a_k \cos(2\pi kt) + b_k \sin(2\pi kt) = \sum_{k=-\infty}^{\infty} c_k e^{2\pi ikt},$$

where a_k and b_k are real coefficients, $e^{i\theta} = \cos(\theta) + i\sin(\theta)$, $c_k = (a_k - ib_k)/2$, and $c_{-k} = \overline{c_k} = (a_k + ib_k)/2$. The coefficients are obtained by

$$a_k = 2\int_0^1 f(t)\cos(2\pi kt)dt \quad \text{and} \quad b_k = 2\int_0^1 f(t)\sin(2\pi kt)dt \qquad (*)$$

or equivalently $c_k = \int_0^1 f(t)\exp(2\pi ikt)dt$. The set of coefficients $\{a_k, b_k : k = 1, 2, \ldots\}$ or equivalently $\{c_k : k = 1, 2, \ldots\}$ is called the *Discrete Fourier Transform (DFT) of* $f(t)$.

Recalling that for any complex number $z = a + ib$, $z\overline{z} = (a + bi)(a - bi) = a^2 + b^2 = |z|^2$, we also know that

$$\int_0^1 f^2(t)dt = \int_0^1 f(t)\overline{f(t)}dt = \int_0^1 f(t) \sum_{k=-\infty}^{\infty} c_k e^{-2\pi ikt}\, dt = \sum_{k=-\infty}^{\infty} c_k\overline{c_k} = \sum_{k=-\infty}^{\infty} |c_k|^2 = 2\sum_{k=1}^{\infty}(a_k^2+b_k^2)/4$$

this is called *Parseval's identity.* The function

$$S(k) = |c_k|^2 = (a_k^2 + b_k^2)/4$$

is called the *spectrum* of $f(t)$; a particular estimate of $S(k)$ from time series data, described below, is called a *periodogram.*

The point: *If $S(k) \approx 0$ for all k beyond some K, we might as well truncate the Fourier series for $f(t)$ to just K terms.*

Fourier Analysis of a Time Series as Linear Regression

This is not quite the way spectral analysis of time series is usually developed, but it will give you the right intuition for the problem.

Consider a very simple linear regression model for a time series y_t exhibiting cyclic variation with known period p, and overall mean $\sum_{t=1}^n y_t/n \approx 0$:

$$y_t = \alpha\cos(\omega t) + \beta\sin(\omega t) + z_t,$$

where z_t is a white noise sequence and $\omega = 2\pi/p$ is the known frequency, i.e. [after some trig identities; try fooling around with $e^{i(a+b)}$] we are fitting a phase-shifted sinusoid

$$\sqrt{\alpha^2+\beta^2}\,\sin(\omega t + t_0)$$

to y_t. This may be rewritten in matrix form as

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} \cos(\omega 1) & \sin(\omega 1) \\ \cos(\omega 2) & \sin(\omega 2) \\ \vdots & \vdots \\ \cos(\omega n) & \sin(\omega n) \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} + \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix}, \quad \text{or} \quad \underline{y} = X\underline{\beta} + \underline{z}$$

We may use the usual matrix formula for the regression coefficients $\hat{\underline{\beta}} = (X'X)^{-1}X'\underline{y}$ together with some trig integrals from Calc II, to show that

$$\begin{aligned} \hat{\alpha} &\approx \frac{2}{n}\sum_{t=1}^n y_t\,\cos(\omega t) \\ \hat{\beta} &\approx \frac{2}{n}\sum_{t=1}^n y_t\,\sin(\omega t) \end{aligned}$$

Clearly, these are discrete approximations to the integrals in $(*)$ defining a one-term DFT for $f(t) = y_{t/p}$.

Moreover, the regression sum of squares is

$$\sum_{t=1}^n \hat{y}_t^2 = \sum_{t=1}^n [\hat{\alpha}\cos(\omega t) + \hat{\beta}\sin(\omega t)]^2 \approx \frac{n}{2}(\hat{\alpha}^2 + \hat{\beta}^2)$$

(a version of Parseval's identity), so that $R^2 = \sum_t \hat{y}_t^2 / \sum_t y_t^2$ is determined by the squared amplitude of the fitted sinusoid.

This is a quite simplistic model. More realistic would be if we not only did not know the amplitude but also didn't know the frequency of the series.

So consider a model of the form

$$y_t = \sum_{k=1}^K \{\alpha_k\cos(\omega_k t) + \beta_k\sin(\omega_k t)\} + z_t;$$

we are now fitting K shifted sinusoids to y_t. This is again an easy model to estimate, provided we pick the right set of frequencies. If we take $\omega_k = 2\pi k/n$, with $k < n/2$ then some nice things happen:

- For starters, *this is the only set of frequencies that we really can estimate*: smaller periods would wiggle more often than the data, longer periods would not see an entire cycle in the data!
- By fooling around with $x = e^{2\pi ikt/n}$ in the identity $(1-x^n) = (1-x)(1+x+x^2+\cdots+x^{n-1})$, one can show that $e^{2\pi ikt/n}$, $k = 1, 2, \ldots$ *represents a set of orthogonal functions.* That is the correlation between any pair of these functions (measured at $t = 1, 2, \ldots, n$) is zero.
- The nicest thing that happens (because of the orthogonality) is that the estimates of $\hat{\alpha}$ and $\hat{\beta}$ are the same as before, namely,

$$\begin{aligned}\hat{\alpha}_k &= \frac{2}{n}\left\{\sum_{t=1}^{n} y_t \cos(\omega_k t)\right\} \\ \hat{\beta}_k &= \frac{2}{n}\left\{\sum_{t=1}^{n} y_t \sin(\omega_k t)\right\}\end{aligned}$$

- Some further manipulations show that

$$\sum_{t=1}^{n}[y_t \cos(\omega_k t) + y_t \sin(\omega_k t)]^2 = \frac{n}{2}(\hat{\alpha}_k^2 + \hat{\beta}_k^2)$$

  so that $\hat{\alpha}_k^2 + \hat{\beta}_k^2$ again gives the sums of squares associated with each of the K fitted sinusoids.
- Finally, (via some math, or common sense), if all the frequencies are fit, then the regression model fits exactly.

The Periodogram

Now we define the *periodogram function* $I(\omega)$ for $0 \leq \omega \leq \pi$, by

$$I(\omega) = \frac{1}{n}\sum_{t=1}^{n}[y_t \cos(\omega t) + y_t \sin(\omega t)]^2.$$

Of course $I(2\pi k/n) = (\hat{\alpha}_k^2 + \hat{\beta}_k^2)/2$ is an estimate of (twice) the spectrum $S(k) = (\hat{\alpha}_k^2 + \hat{\beta}_k^2)/4$ of the time series, and is proportional to the squared amplitude of the sinusoid

$$\sqrt{\alpha_k^2 + \beta_k^2}\,\sin(\omega_k t + t_k)$$

fitted at frequency $\omega_k = 2\pi k/n$, and represents that power (amount of energy) that can be estimated at each of these fundamental frequencies.

Notice that we can "parition" the total sum of squares for the data y_t as

$$\sum_{t=1}^{n} y_t^2 = nI(0) + 2n\sum_{k=1}^{K_{max}} I(2\pi k/n)$$

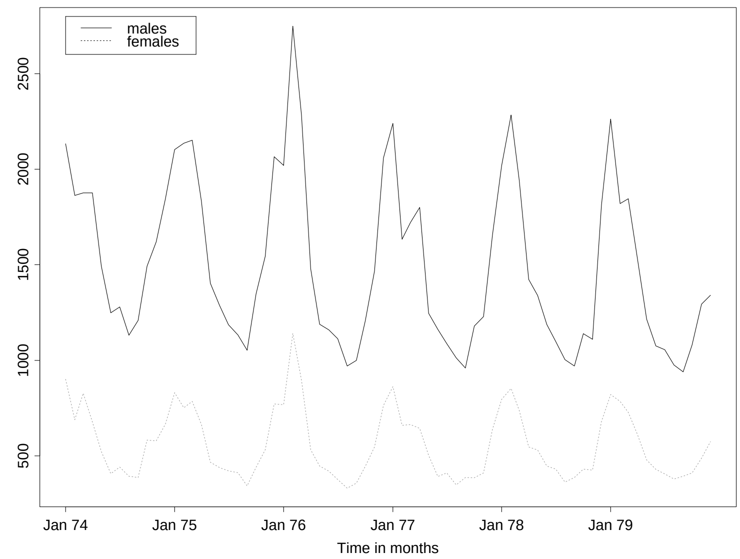
where $I(0)$ corresponds to the intercept $\sum_{t=1}^{n} y_t$ if it is nonzero, and $K_{max} = \lfloor n/2 \rfloor$ is the largest integer less than or equal to $n/2$. On the one hand, this is just Parseval's identity again. On the other, it suggests the following ANOVA table, for $1 \leq K \leq K_{max}$, when $\sum y_t = 0$:

k		SS	*Periodogram*
1	$SS_1 =$	$\sum_{t=1}^n [y_t \cos(\omega_1 t) + y_t \sin(\omega_1 t)]^2$	$= 2n\, I(2\pi/n)$
2	$SS_2 =$	$\sum_{t=1}^n [y_t \cos(\omega_2 t) + y_t \sin(\omega_2 t)]^2$	$= 2n\, I(2\pi 2/n)$
$\vdots$	$\vdots$		$\vdots$
K	$SS_K =$	$\sum_{t=1}^n [y_t \cos(\omega_K t) + y_t \sin(\omega_K t)]^2$	$= 2n\, I(2\pi K/n)$
Regression	$SSR =$	$\sum_{t=1}^n \hat{y}_t^2 \; = \; \sum_{k=1}^K SS_k$	$= 2n \sum_{k=1}^K I(2\pi k/n)$
Error	$SSE =$	$\sum_1^n (y_t - \hat{y}_t)^2 \; = \; \sum_{k=K+1}^{n/2} SS_k$	$= 2n \sum_{k=K+1}^{n/2} I(2\pi k/n)$
Total	$SST =$	$\sum_1^n y_t^2 \; = \; \sum_1^n \hat{y}_t^2 + \sum_1^n (y_t - \hat{y}_t)^2$	$= 2n \sum_{k=1}^K I(2\pi k/n)$

If you go back to your old linear regression notes, and examine the matrix form of regression $\underline{y} = X\underline{\beta} + \underline{\varepsilon}$ when the columns of X are orthogonal, you will see that this is the same ANOVA table: each SS is the dot product of $\underline{y}$ with the corresponding column(s) of X, and (because of orthogonality) it doesn't matter what order the independent variables are entered in the regression.

Some Examples

Male and Female Respiratory Deaths in the UK. First, we re-plot the male and female deaths in the UK from various respiratory illnesses.



In the male sequence, there appears to be a clear 12-month cycle, and a hint of a cycle at about five or six months. Not much else is evident from these plots.

The DFT of a discrete sequence y_t can be conveniently computed in SPLUS with the so-called "fast Fourier transform", `fft()`. Thus, a *really* raw periodogram may be calculated as follows (the `Mod` function computes the modulus of a complex number):

```
402 > par(mfrow=c(4,1))
402 > length(uk.male)
[1] 72
402 > plot((1:35)/6,10/log(10)*log(Mod(fft(uk.male)))[2:36],type="l")
402 > title("Really raw periodogram")
```

The values of the periodogram $I(\omega_k)$ have been plotted on a decibel scale, $10 \cdot \log_{10} S(\omega_k)$, since that is how the periodogram is usually plotted.

The overall mean is not zero here, so we should take it out before we compute the periodogram. Often, not only a mean but a linear trend is taken out of the data before calculating the periodogram, for example:

```
402 > uk.resids _ residuals(lm(resp ~ time, data.frame(time=1:72,resp=c(uk.male))))
402 > plot((1:35)/6,10/log(10)*log(Mod(fft(uk.resids)))[2:36],type="l")
402 > title("Really raw periodogram, detrended")
```

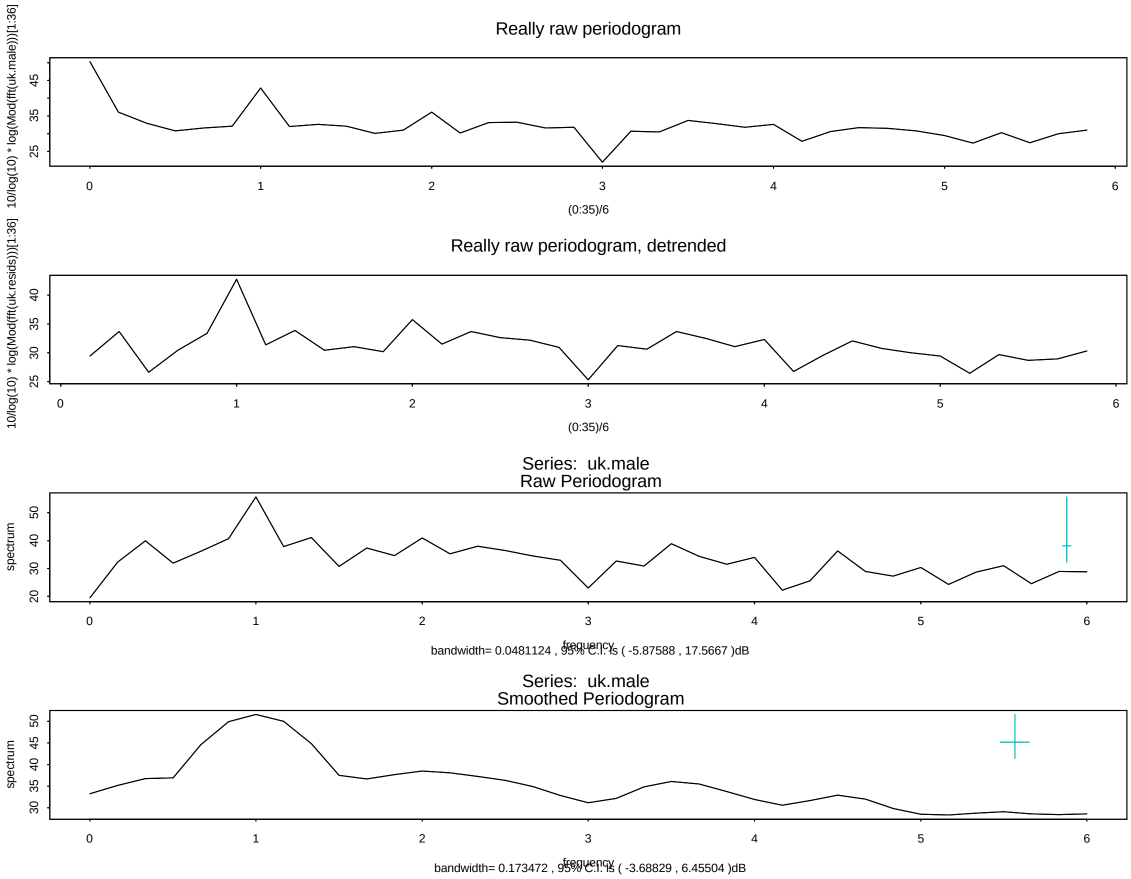
The function `spectrum` in SPLUS also computes a periodogram from the detrended time series, but in addition it smooths the observations at the beginning and end of the series; and allows us to run one or more moving-average smooths over the final periodogram.

```
402 > spectrum(uk.male)
402 > spectrum(uk.male,spans=c(3,3))
402 > # smooth with running means of length 3, twice
```

Graphs of these various periodogram estimates follow. By convention, all results are plotted on a decibel ($10 \log_{10}$) scale.

Notice that the main power corresponds to a 12 months cycle, but there is also significant power at some other frequencies (a half- or one-third-year cycle and a two-year cycle). Remember, this is like looking at an ANOVA table to decide which terms to keep in the regression.

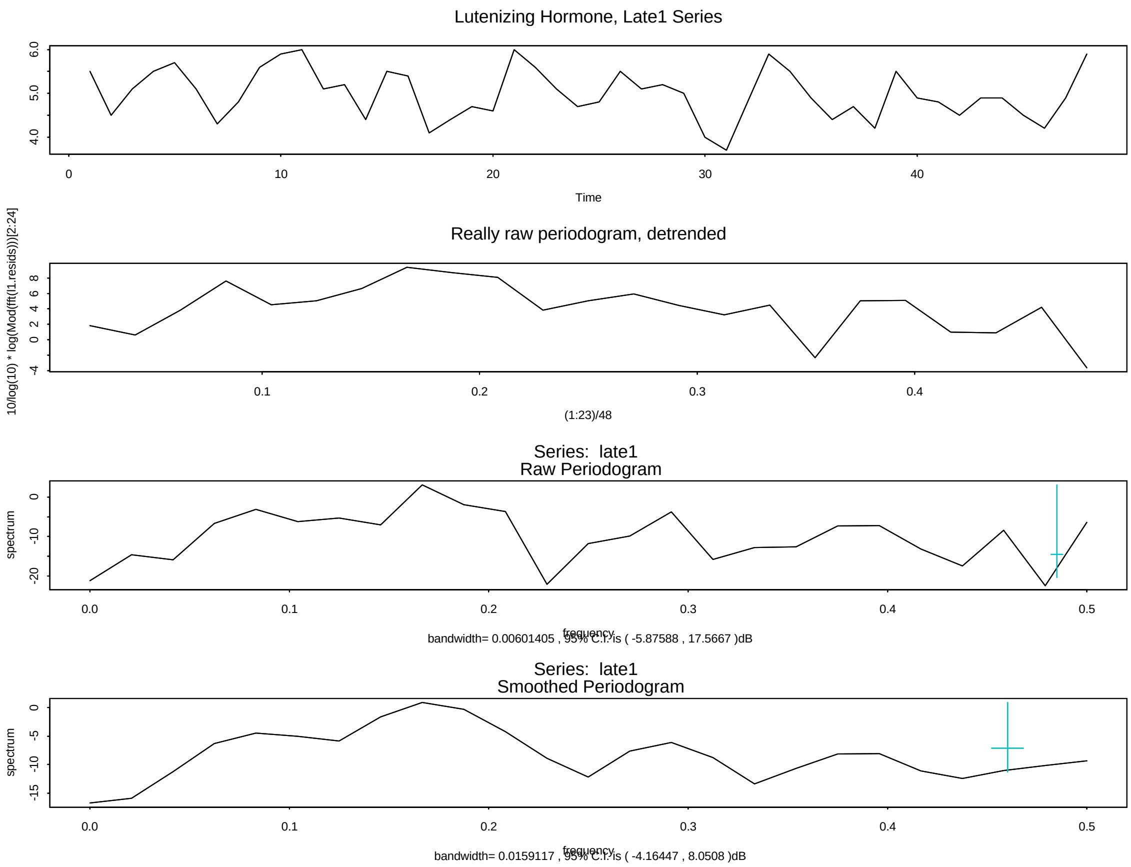
Raw and smoothed periodograms of the `uk.female` series are essentially indistinguishable from these.



Lutenizing hormone. We will look at the `late1` series from these data.

```
402 > ts.plot(late1)
402 > title("Lutenizing Hormone, Late1 Series")
402 > length(late1)
[1] 48
402 > l1.resids _ residuals(lm(resp~time,data.frame(resp=c(late1),time=1:48)))
402 > plot((1:23)/48,10/log(10)*log(Mod(fft(l1.resids)))[2:24],type="l")
402 > title("Really raw periodogram, detrended")
402 > spectrum(late1)
402 > spectrum(late1,span=3) # follow with a single running mean of length 3
```

The plots below suggest there is a lot of power at three wavelengths, 8 = .17*48, 14 = .29*48 and 19 = .40*48, although there appears to be "junk" at shorter wavelengths as well.



Serial Dependence in Time Series

The second popular way to analyze time series data is "state space analysis" or "looking at the time series in the space domain". This method concentrates on the serial dependence rather than the cyclic behavior of the series. The most widely used models are ARMA and ARIMA processes. We introduce the basic AR and MA processes here.

Autocovariance and autocorrelation function

Recall that the *autocovariance function* of a stationary random function $\{Y(t)\}$ is

$$\gamma(k) = \text{Cov}\{Y(t), Y(t-k)\}$$

This does not not depend on t because of the the assumption of stationarity. Since $\gamma(0)$ is the variance, the *autocorrelation function* is

$$\rho(k) = \gamma(k)/\gamma(0)$$

Basic time series models: AR and MA

The two basic building-blocks for the "noise" part of a time series model are the autoregressive (AR) and moving average (MA) processes. MA processes are easier to describe first.

- A *moving average* process of order q, denotes $MA(q)$, has the form

$$y_t = \mu + \beta_0 z_t + \beta_1 z_{t-1} + \ldots + \beta_q z_{t-q}$$

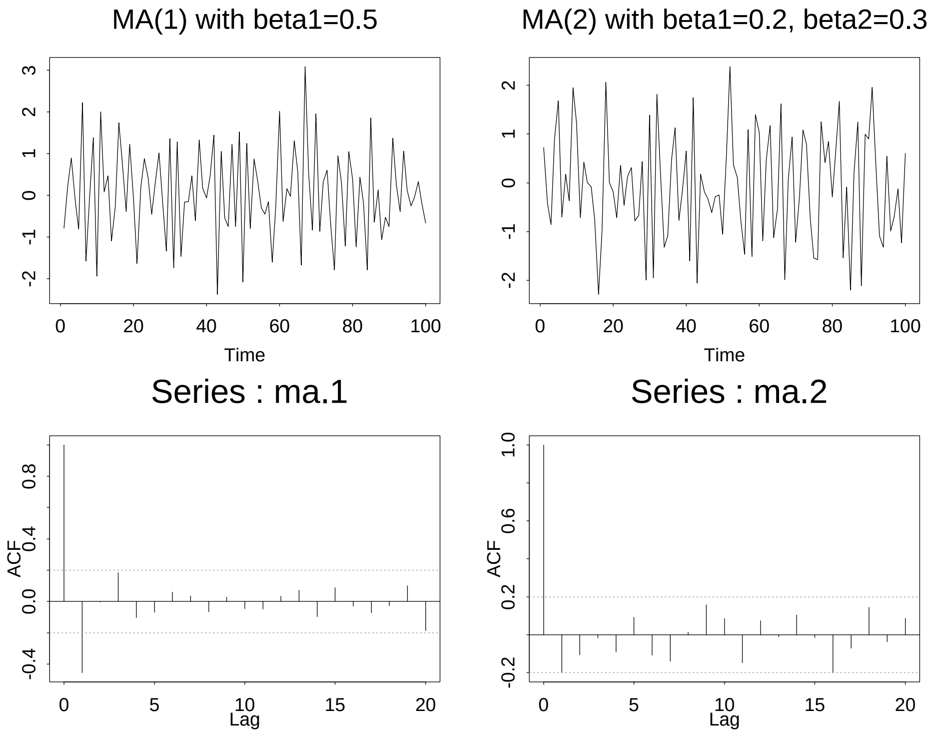
or, in terms of the *backshift operator* $Bz_t = z_{t-1}$,

$$\begin{aligned} y_t - \mu &= (\beta_0 + \beta_1 B + \ldots \beta_q B^q) z_t \\ &\equiv \Theta(B) z_t, \end{aligned}$$

where z_t is a mean-zero white noise process. It is easy to show that y_t is always stationary, and

$$\gamma(k) = \begin{cases} \sum_{t=0}^{q-|k|} \beta_t \beta_{t+|k|} & |k| \le q \\ 0 & |k| > q \end{cases}$$

Here are plots of an $MA(1)$ and an $MA(2)$ process.



The autocorrelation plot for a $MA(q)$ process drops to zero (except for noise) after lag q, which make the `acf` plot a good first step in guessing the order q of the process.

- An *autoregressive* process of order p, denoted $AR(p)$ has the form

$$y_t = \alpha_1 y_{t-1} + \alpha_2 y_{t-2} + \cdots + \alpha_p y_{t-p} + z_t$$

where again z_t is a mean-zero white noise process. In terms of the *backshift operator*, $By_t = y_{t-1}$, we may write

$$(1 - \alpha_1 B - \alpha_2 B^2 - \cdots - \alpha_p B^p) y_t = z_t.$$

If $E[y_t] = \mu \neq 0$, then we would write

$$(1 - \alpha_1 B - \alpha_2 B^2 - \cdots - \alpha_p B^p)(y_t - \mu) = z_t.$$

Not all values of the coefficients α_k lead to a stationary series. For example we saw last time that the random walk $y_t = y_{t-1} + z_t$ was *not* a stationary process. More generally the complex roots of the polynomial

$$\Phi(x) = 1 - \alpha_1 x - \alpha_2 x^2 - \cdots - \alpha_p x^p$$

must have all p of its roots outside the unit circle in the complex plane. In the case of an AR(1) process, this is easy to show!

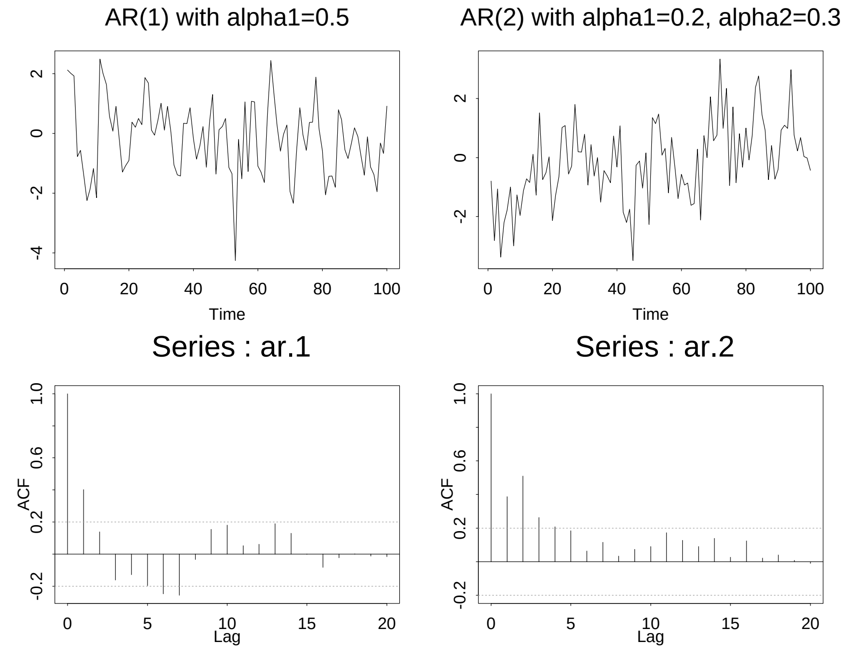
One can show that the autocovariance function for a stationary $AR(p)$ process satisfies the "Yule-Walker" relations

$$\sum_{k=1}^{p} \gamma(k-i)\alpha_k = \gamma(k) \qquad i = 1, 2, \ldots, p,$$

We can use the Yule-Walker equations to solve for $\gamma(k)$ in terms of the α's, if they are known, or solve for the α's in terms of the $\gamma(k)$'s. This suggests a simple way to estimate the coefficients of an AR process. Also,

$$\sigma_y^2 = \sum_{k=1}^{p} \gamma(k)\alpha_k + \sigma_z^2.$$

As an exercise, show that the autocorrelation function of an $AR(1)$ process $y_t = \alpha y_{t-1} + z_t$ satisfies $\rho_k = \alpha \rho_{k-1}$, and so, inserting $\rho_0 = 1$, we get $\rho_k = \alpha^k$. What does ρ_k look like for an $AR(2)$ process? Here are plots of simulated $AR(1)$ and $AR(2)$ processes.



The `acf` plot of an $AR(1)$ decays exponentially to zero, possibly with some oscillation in sign. The $AR(2)$ `acf` plot also decays to zero (this gradual decay in the `acf` plot distinguishes pure-AR from pure-MA processes) but it isn't easy to tell from the plot what the order p of the process should be.

Partial Autocorrelation Function

Here are some more examples of AR processes and their acf's:

- The following graph shows plots of two AR processes,

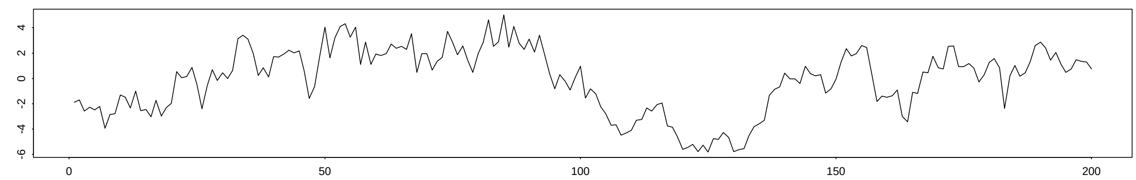
$$y_t = .7y_{t-1} + .2y_{t-2} + z_t$$

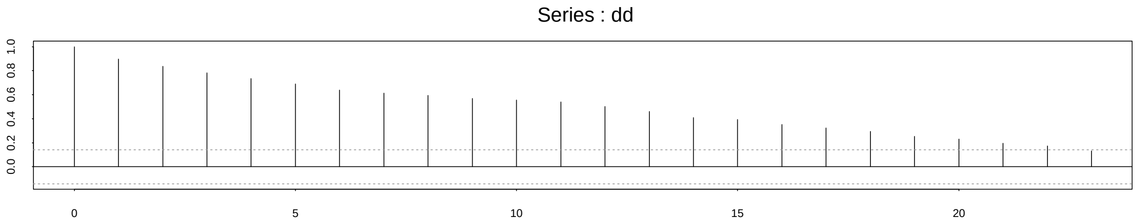
and

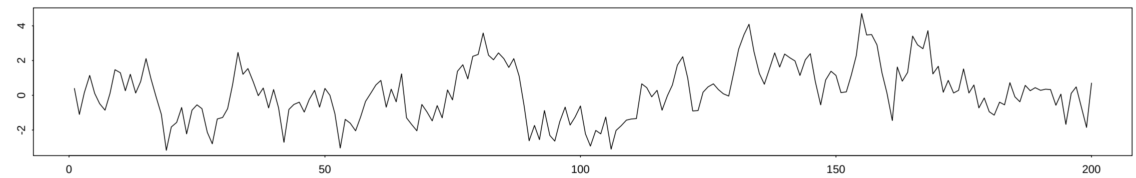
$$y_t = .7y_{t-1} + z_t$$

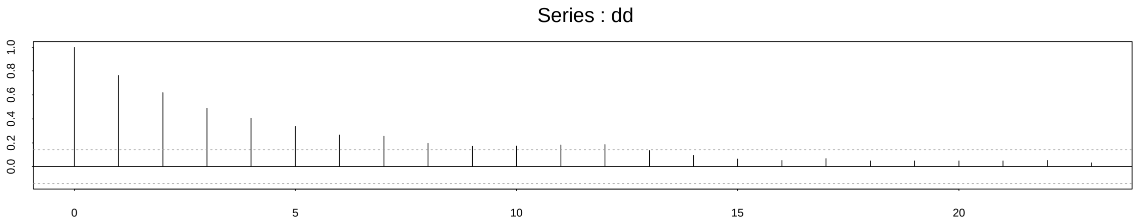
and their autocorrelation functions. Notice that the ACF's have long lags and it is not easy to diagnose anything.

```
402 > tsplot(dd <- arima.sim(200, model=list(ar=c(.7, .2))))
402 > acf(dd)
402 > tsplot(dd <- arima.sim(200, model=list(ar=c(.7))))
402 > acf(dd)
```



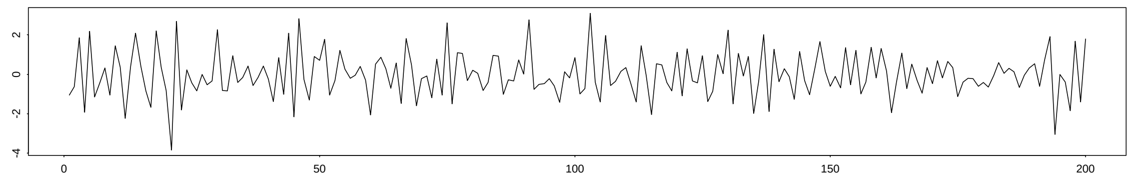


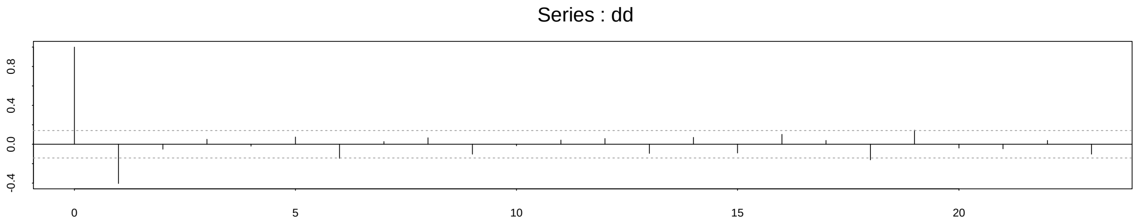


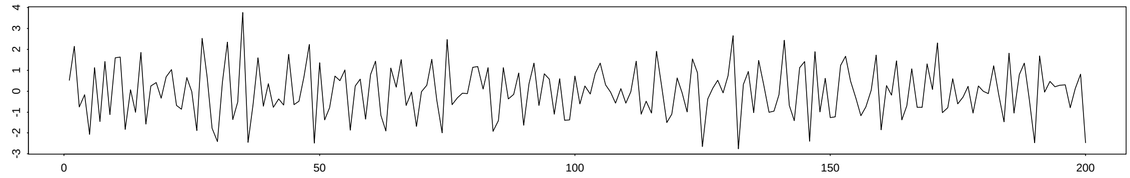


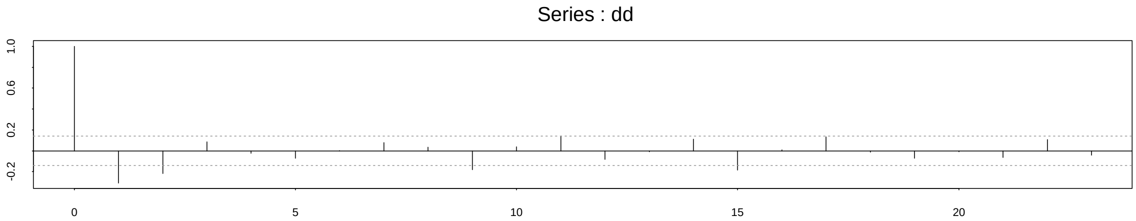
- Next we plot two more MA processes and their acf's:

```
402 > tsplot(dd <- arima.sim(200, model=list(ma=c(.6))))
402 > acf(dd)
402 > tsplot(dd <- arima.sim(200, model=list(ma=c(.6, .4))))
402 > acf(dd)
```









The two models that we fit are

$$y_t = .6z_{t-1} + z_t$$

and

$$y_t = .6z_{t-1} + .4z_{t-2} + z_t$$

Just as before, that the ACF tails off after lag 1 and 2. That is the ACF provides a diagnostic of the order of the MA process.

A simple (too simple!) conclusion is that if the ACF is zero beyond a certain lag, then that is evidence for an MA process of the appropriate order. That turns out to be a useful observation, but it doesn't help us with AR processes.

The form of an autoregressive model,

$$y_t = \alpha_1 y_{t-1} + \alpha_2 y_{t-2} + \cdots + \alpha_p y_{t-p} + z_t$$

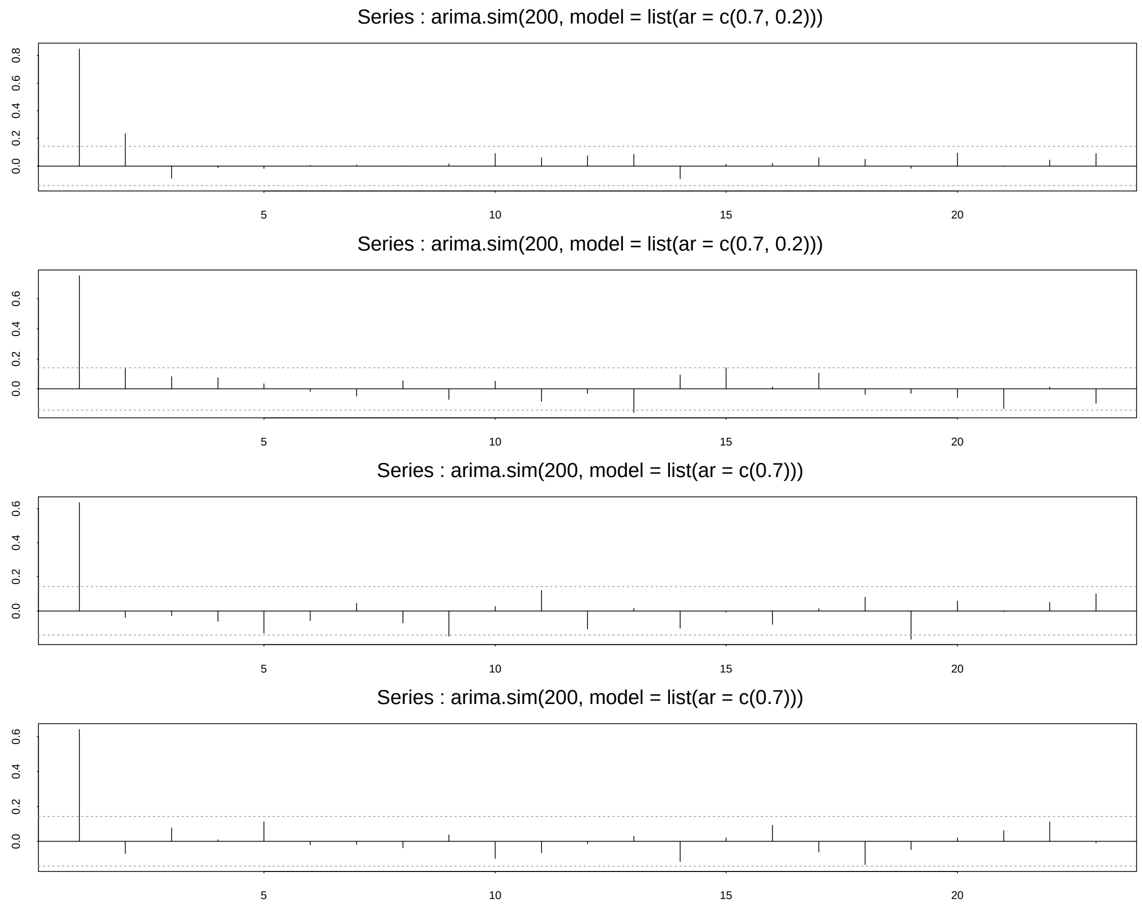
suggests that instead of plotting the covariances between lagged observations, we should be plotting the coefficients α_k themselves: essentially, we want to fit AR models of orders 1, 2, 3, and so on. Then at lag 1 we plot the coefficient α_1 of the fitted AR(1) model. At lag 2 we plot the coefficient α_2 of the AR(2) model, and so on. When p is too high, the plotted coefficients should be essentially zero. This type of plot is called a partial autocorrelation function (PACF), and it works really well.

To actually calculate the PACF (and plot it in SPLUS) requires the use of the acf function with an extra argument.

```
acf(dd,type="p")
```

I have produced four plots, two of AR(2) processes and two of AR(1) processes. Notice that these plots look like the ACF plots of MA processes.

```
402 > acf(arima.sim(200, model=list(ar=c(.7, .2))), type="p")
402 > acf(arima.sim(200, model=list(ar=c(.7, .2))), type="p")
402 > acf(arima.sim(200, model=list(ar=c(.7))), type="p")
402 > acf(arima.sim(200, model=list(ar=c(.7))), type="p")
```

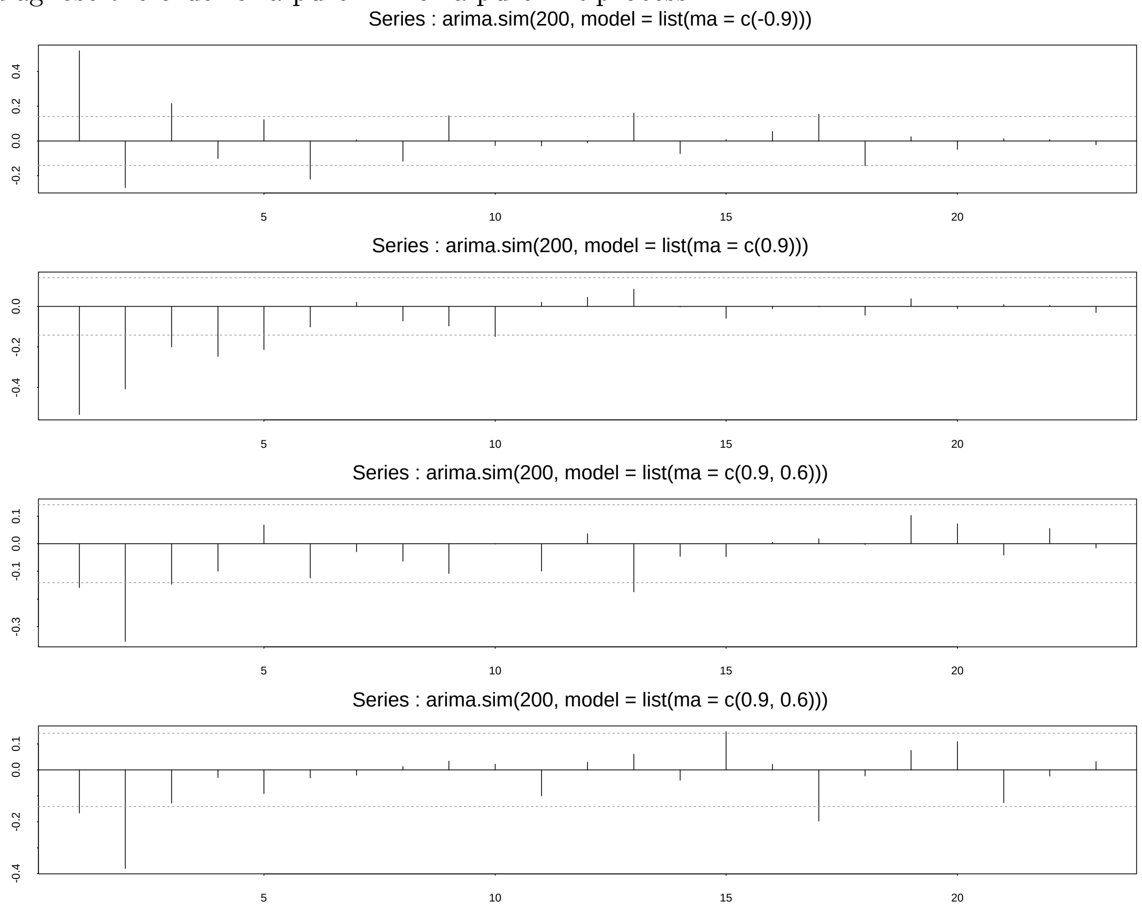


Maybe the PACF of a MA process will look like the ACF of an AR process? This sort of duality would be quite useful. Lets look and see. The simulations were generated by

```
402 > acf( arima.sim(200, model=list(ma=c(-.9))), type="p")
```

```
402 > acf( arima.sim(200, model=list(ma=c(.9))), type="p")
402 > acf( arima.sim(200, model=list(ma=c(.9, .6))), type="p")
402 > acf( arima.sim(200, model=list(ma=c(.9, .6))), type="p")
```

The following plots really do look like the ACF of AR's. There is a mathematical duality between ACF's and PACF's, but the important thing to note is that each of them is quite good at helping to diagnose the order of a pure MA or a pure AR process.



Thus,

- To get an idea what the order q of a pure $MA(q)$ process should be, use an ACF plot.
- To get an idea what the order p of a pure $AR(p)$ process should be, use a PACF plot.

There are better ways to choose p and q, but these make useful illustrations and can help you diagnose whether you are "way off".