14.384 Time Series Analysis, Fall 2007
Recitation by Paul Schrimpf
Supplementary to lectures given by Anna Mikusheva
September 11, 2008

Recitation 2: Time Series in Matlab

# Time Series in Matlab

In problem set 1, you need to estimate spectral densities and apply common filters. You can use any software you would like, but we recommend using Matlab. It may be easier to do simple things using more statistics oriented programs like Stata or RATs, since these programs include pre-packaged commands for many common tasks, but you will learn more by writing the code yourself. Also, it is generally easier to write programs for new estimators in a full-featured programming language like Matlab's than in the language of statistics oriented programs. Finally, I recommend using Matlab because I happen to use Matlab, and I will be more likely to be able to provide help if you need it.

These notes cover some slightly obscure Matlab commands that can be useful for time series. For a more general overview, see
http://web.mit.edu/~paul_s/www/14.170/matlab.html

*Disclaimer: I wrote these notes last year, and I am not entirely sure that they are completely correct. Many of the commands covered are from Matlab's signal processing toolbox, and they have different names and may do slightly different things than what an econometrician would expect. Always read Matlab's help and documentation before using a command. When in doubt, double check that the command does what you think.*

## Simulating an ARMA Model

$a(L)y_t = b(L)e_t$

```
clear;
a = [1 0.5]; % AR coeffs
b = [1 0.4 0.3]; % MA coeffs
T = 1000;
e = randn(T,1); % generate gaussian white noise
y = filter(b,a,e); % generate y
```

The `filter` function can be used to generate data from an ARMA model, or apply a filter to a series.

## Impulse-Response

To graph the impulse response of an ARMA, use `fvtool`

```
% create an impulse response
fvtool(b,a,'impulse');
```

## Sample Covariances

```
[c lags]=xcov(y,'biased');
figure;
plot(lags,c);
title('Sample Covariances');
```

The option, `'biased'`, says to use $\hat{\gamma}_k = \frac{1}{T}\sum_{t=1}^{T-k}(y_t - \bar{y})(y_{t-k} - \bar{y})$; `'unbiased'` would use $\hat{\gamma}_k = \frac{1}{T-k}\sum_{t=1}^{T-k}(y_t - \bar{y})(y_{t-k} - \bar{y})$

## Spectral Analysis

Population spectral density for an ARMA:

```
% population density
w = 0:0.1:pi;      % frequencies to compute density at
h = freqz(b,a,w);  % returns frequency response = b(e^{iw})/a(e^{iw})
sd = abs(h).^2./sqrt(2*pi); % make into density
```

### Estimating the Spectral Density

**Parametric methods** These estimate an $AR(p)$ and use it to compute the spectrum.

```
[sdc wc] = pcov(y,8); % estimate spectral density by fitting AR(8)

[sdy wy] = pyulear(y,8); % estimate spectral density by fitting AR(8)
                         % using
                         % the Yule-walker equations
```

**Non-parametric methods**

**Definition 1.** The *sample periodogram* of $\{x_t\}_{t=1}^{T}$ is $\hat{S}(\omega) = \frac{1}{T}|\sum_{t=1}^{T} e^{-i\omega t}x_t|^2$

*Remark* 2. The sample periodogram is equal to the Fourier transform of the sample autocovariances

$$\hat{S}(\omega) = \frac{1}{T}|\sum_{t=1}^{T} e^{-i\omega t} x_t|^2 = \sum_{k=-T-1}^{T-1} \hat{\gamma}_k e^{-i\omega k}$$

```
[sdp wp] = periodogram(y,[],'onesided'); % estimate  using sample
```

**Definition 3.** A *smoothed periodogram* estimate of the spectral density is

$$\hat{S}(\omega) = \int_{-\pi}^{\pi} h_T(\lambda - \omega) \frac{1}{T}|\sum_{t=1}^{T} e^{-i\lambda t} x_t|^2 d\lambda$$

where $h_T()$ is some kernel weighting function.

A smoothed periodogram is a weighting moving average of the sample periodogram.

The following code estimates a smoothed periodogram using a Parzen kernel with bandwidth $\sqrt{T}$.

```
rT = round(sqrt(T));
[sdw ww] = pwelch(y,parzenwin(rT),rT-1,[],'onesided'); % smoothed periodogram
```

**Definition 4.** A *weighted covariance* estimate of the spectrum is:

$$\hat{S}(\omega) = \sum_{k=-S_T}^{S_T} \hat{\gamma}_k g_T(k) e^{-i\omega k}$$

where $g_T(k)$ is some kernel.

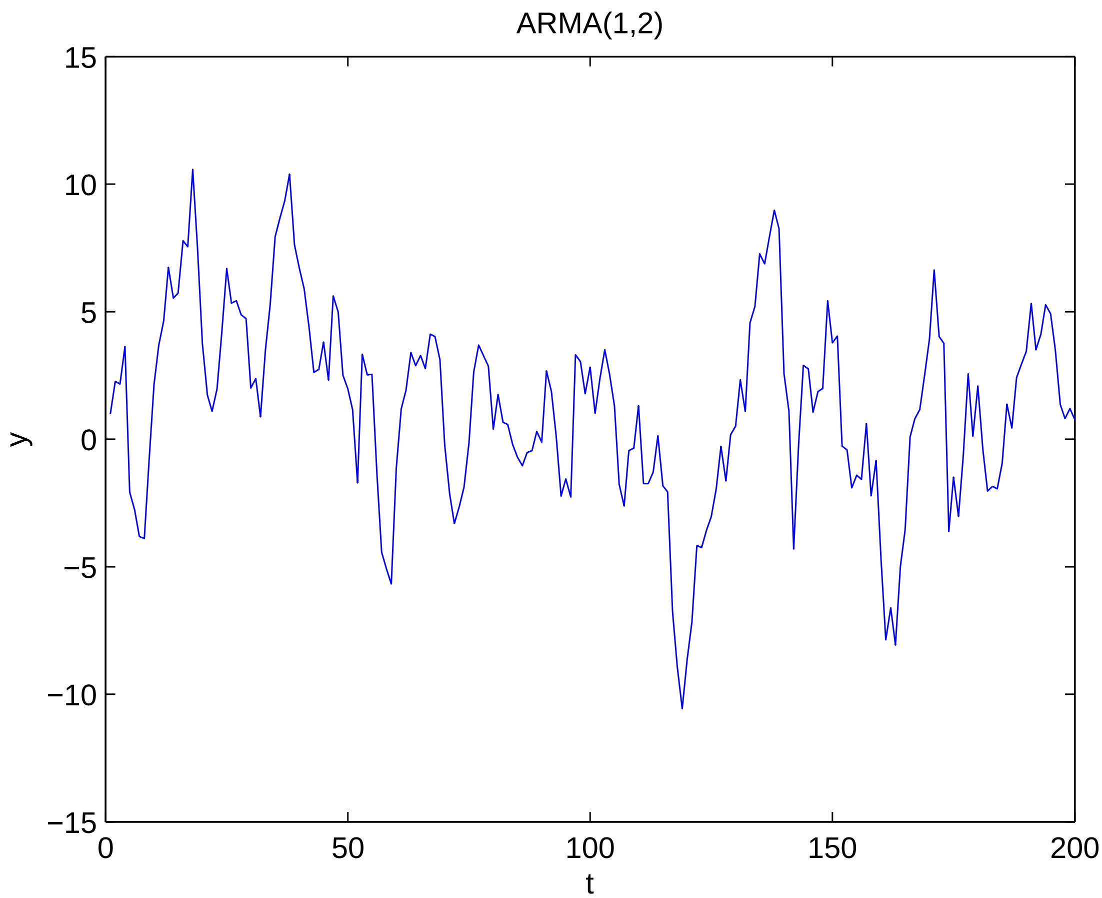
```
% bartlett weighted covariances
wb = 0:0.1:pi;
rT = sqrt(T);
[c t]=xcov(y,'biased');
weight = 1-abs(t)/rT;
weight(abs(t)>rT) = 0;
for j=1:length(wb)
  sdb(j) = sum(c.*weight.*exp(-i*wb(j)*t));
end
sdb = sdb / sqrt(2*pi);
```

## Filtering

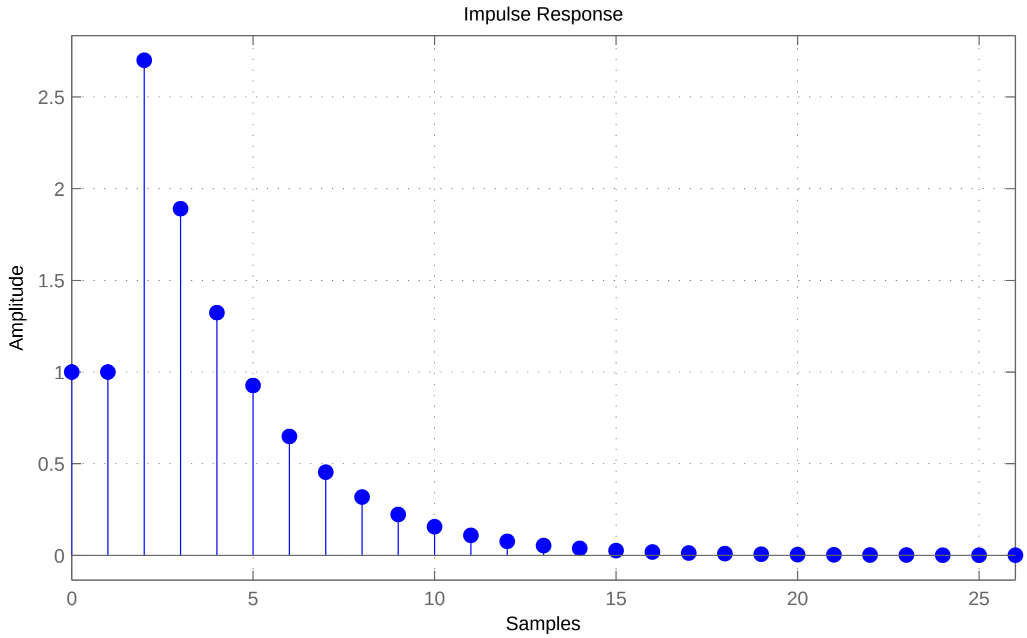
See: http://ideas.repec.org/c/wpa/wuwppr/0507001.html for code for common filters.

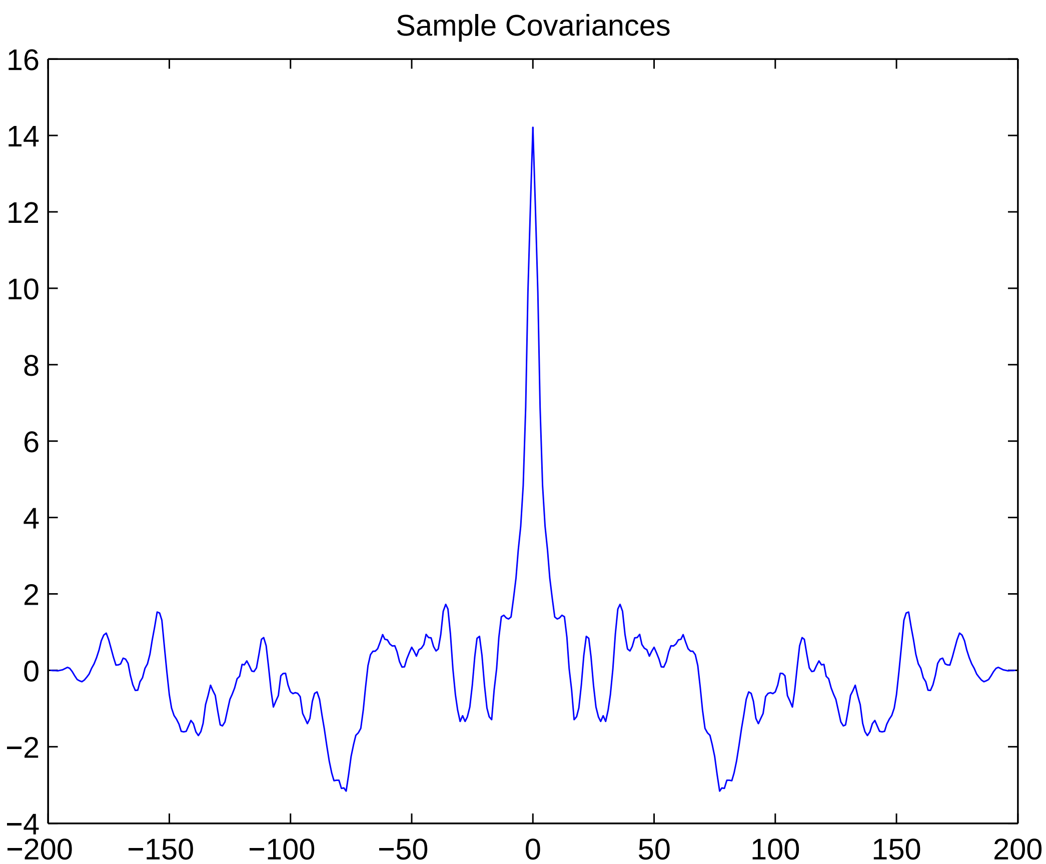
# Example: Simulating an ARMA and estimating the spectrum

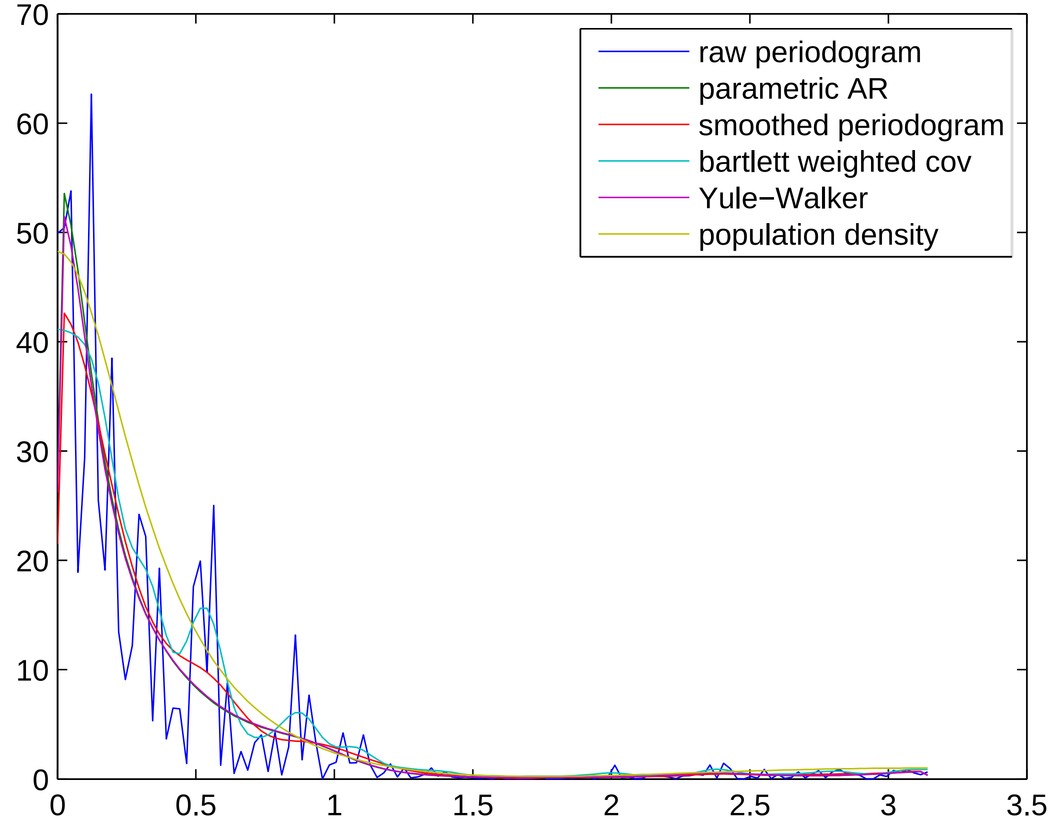
```
clear;
close all; % closes all open figure windows

% model: y_t = 0.9 y_{t-1} + b(L) e_t
a = [1 -0.7]; % AR coeffs
b = [1 0.3 2]; % MA coeffs
T = 200;
e = randn(T,1); % generate gaussian white noise
y = filter(b,a,e); % generate y

% plot y
figure;
plot(y);
xlabel('t');
ylabel('y');
title('ARMA(1,2)');

% create an impulse response
fvtool(b,a,'impulse');

% calculate and plot sample auto-covariances
[c lags]=xcov(y,'biased');
figure;
plot(lags,c);
title('Sample Covariances');

% estimate spectral density

% parametric
[sdc wc] = pcov(y,8); % estimate spectral density by fitting AR(8)
[sdy wy] = pyulear(y,8); % estimate spectral density by fitting AR(8)
                         % using
                         % the Yule-walker equations

% nonparametric
[sdp wp] = periodogram(y,[],'onesided'); % estimate  using unsmoothed
                                         % periodogram
rT = round(sqrt(T))*3;
[sdw ww] = pwelch(y,parzenwin(rT),rT-1,[],'onesided'); % smoothed
                                         periodogram

% bartlett weighted covariances
[c lags]=xcov(y,'biased');
t = -(T-1):(T-1);
weight = 1-abs(t')/rT;
weight(abs(t')>rT) = 0;
wb = ww;
for j=1:length(wb)
```




```
  sdb(j) = sum(c.*weight.*exp(-i*wb(j)*(-(T-1):(T-1))'));
end
sdb = sdb / sqrt(2*pi);

% population density
w = wb;
h = freqz(b,a,w);
sd = abs(h).^2./sqrt(2*pi);

figure;
plot(wp,sdp,wc,sdc,ww,sdw,wb,sdb,wy,sdy,w,sd);
legend('raw periodogram','parametric AR','smoothed periodogram', ...
       'bartlett weighted cov','Yule-Walker','population density');
```


Impulse Response
Amplitude
2.5
2
1.5
1
0.5
0
0
5
10
15
20
25
Samples


Sample Covariances
16
14
12
10
8
6
4
2
0
−2
−4
−200
−150
−100
−50
0
50
100
150
200


70
60
50
40
30
20
10
0
0
0.5
1
1.5
2
2.5
3
3.5
raw periodogram
parametric AR
smoothed periodogram
bartlett weighted cov
Yule−Walker
population density

14.384 Time Series Analysis
Fall 2013