Macroeconometrics - Problem I

Jérôme Romano

October 31, 2023

Exercise 1:

From FREDII data base (http://research.stlouisfed.org/fred2/) download the series GDPC1 (quarterly US real GDP).

Transform the series in growth rates. Suppose that real GDP growth rates follow an AR(1) model:

$$y_t = c + \phi y_{t-1} + \varepsilon_t$$

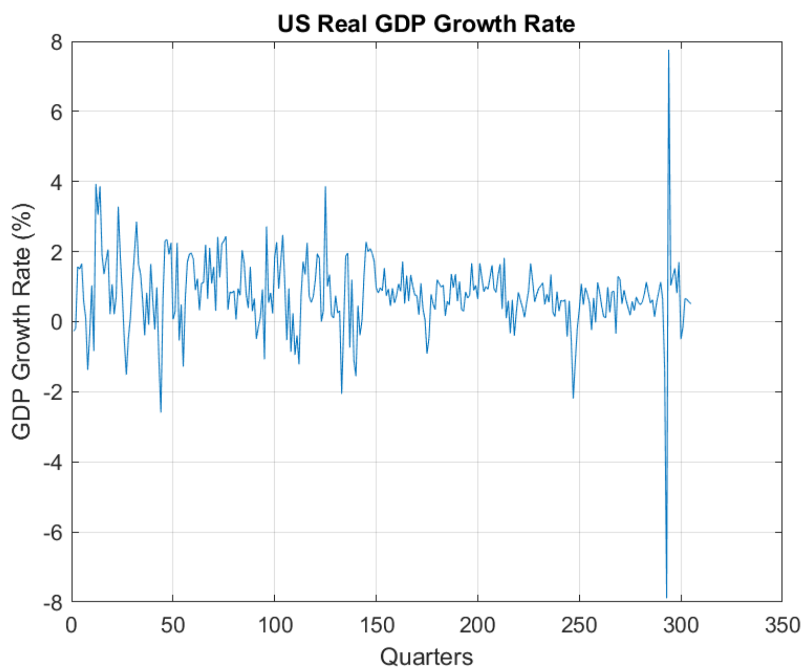
where $\varepsilon_t \sim \text{iid}(0, \sigma^2)$.

(a) Plot the series.

In order to plot the series that contains the growth rate in quarterly GDP we have to create the variable that stores the growth rates starting from the GDP values. The MATLAB code is provided:

```
GDPC1 = readtable('GDPC1.csv')
%% (a) Plot the series

% we import manually the data from FREDII

% Extract the GDP values
gdp_values = GDPC1.GDPC1 ;

% Calculate the growth rates
gdp_growth = diff(gdp_values) ./ gdp_values(1:end-1) *
    100;

% Create an array of quarters for plotting
quarters = 1:numel(gdp_growth);
```

```
% Plot the GDP growth rates
plot(quarters, gdp_growth);

% Add labels and title to the plot
xlabel('Quarters');
ylabel('GDP Growth Rate (%)');
title('US Real GDP Growth Rate');

% Add a grid for better visualization
grid on;
```



(b) Estimate the parameters c and ϕ with OLS.

For the estimation of c and $\hat{\phi}$ with OLS, we create a generic function that estimates, through OLS, the parameter $\hat{c}$ and the vector of $\hat{\phi}$ for a generic Autoregressive Process of order p, i.e., AR(p). The following MATLAB code displays the function called `ar_estimation(p, data)`:

```
% Lag the GDP growth data by one and two quarters -
    this is done in order
```

```
% to achieve harmonization in the length of the matrix
lag1_gdp_growth = lagmatrix(gdp_growth, 1);
lag2_gdp_growth = lagmatrix(gdp_growth, 2);

% Extract lagged values of GDP growth (excluding the
   first two rows)
y_tminus2 = gdp_growth(3:end);      % y(t-2)
y_tminus1 = lag1_gdp_growth(3:end); % y(t-1)
y_t = lag2_gdp_growth(3:end);       % y(t)

% Create the independent variable matrix 'x' using
   lagged values
x = [y_tminus1, y_tminus2];

% Fit a linear regression model using the lagged
   values
model = fitlm(x, y_t);

% Estimate an AR(1) model using the 'ar_estimation'
   function
[c, phi] = ar_estimation(1, gdp_growth);

>> disp('Value c:');
disp(c);
disp('Value phi:');
disp(phi);
Value c:
    0.6701

Value phi:
    0.1341
```

(c) Estimate and plot the first 10 autocorrelations.
Remember that in an AR(1) process defined by the following equation:

$$y_t = c + \phi y_{t-1} + \varepsilon_t \quad (2)$$

the autocorrelation function can be calculated in terms of the covariance between y_t and y_{t-k} (where k is the lag):

$$\rho(k) = \frac{\mathrm{Cov}(y_t, y_{t-k})}{\mathrm{Var}(y_t)} \quad (3)$$

The covariance between y_t and y_{t-k} can be calculated using the definition of an AR(1) process and the properties of white noise errors. Assuming that white noise errors are uncorrelated over time, the covariance is:

$$\mathrm{Cov}(y_t, y_{t-k}) = \phi^k \mathrm{Var}(y_t) \quad (4)$$

Now you can substitute the calculated covariance into the formula for the autocorrelation function:

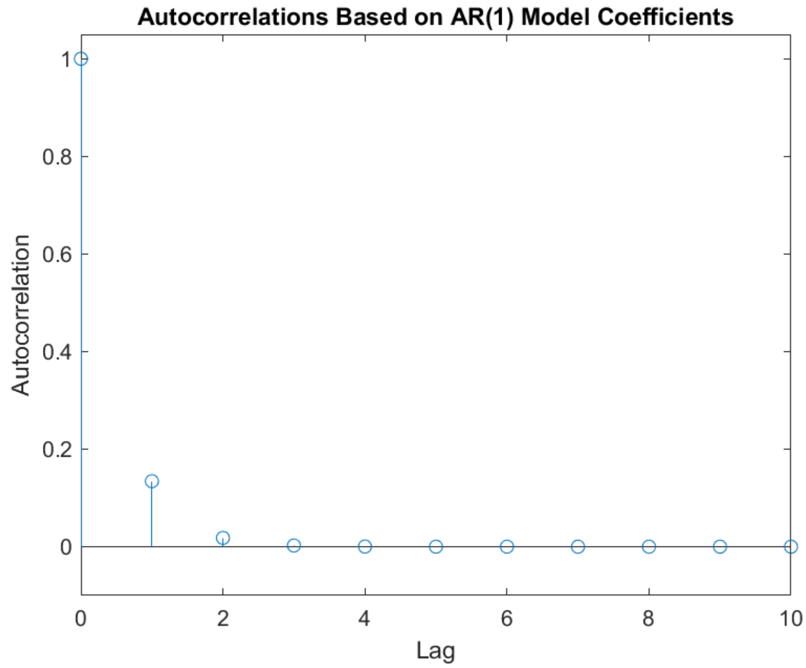
$$\rho(k) = \frac{\phi^k \mathrm{Var}(y_t)}{\mathrm{Var}(y_t)} \quad (5)$$

Finally, simplify the formula:

$$\rho(k) = \phi^k \quad (6)$$

The following MATLAB code evaluates the autocorrelation function:

```
% Define the number of lags
maxLag = 10;

% Preallocate an array to store autocorrelation values
autocorrelations = zeros(1, maxLag + 1);

% Calculate the autocorrelation values based on the AR
    (1) model
autocorrelations(1) = 1; % Autocorrelation at lag 0 is
     always 1
for lag = 1:maxLag
    autocorrelations(lag + 1) = phi ^ lag;
end

% Plot the autocorrelations
figure;
stem(0:maxLag, autocorrelations);
xlabel('Lag');
ylabel('Autocorrelation');
ylim([min(-0.1, min(autocorrelations) - 0.05), max
    (0.2, max(autocorrelations) + 0.05)]);
title('Autocorrelations Based on AR(1) Model
    Coefficients');

% Display the autocorrelation values
fprintf('Autocorrelation values for lags 0 to %d:\n',
    maxLag);
for lag = 0:maxLag
    fprintf('Lag %d: %f\n', lag, autocorrelations(lag
        + 1));
end
```



(d) Obtain and plot the coefficients of the Wold representation of y_t.

In order to obtain the Wold Decomposition, we have to rewrite our process in the form of

$$y_t = \sum_{j=0}^{\infty} \theta_j \varepsilon_{t-j} + \eta_t,$$

i.e., in terms of the series' shocks. We can write:

$$y_t = c + \phi y_{t-1} + \varepsilon_t,$$

substituting y_{t-1} with its expression:

$$y_t = c + \phi(c + \phi y_{t-2} + \varepsilon_{t-1}) + \varepsilon_t \quad (7)$$

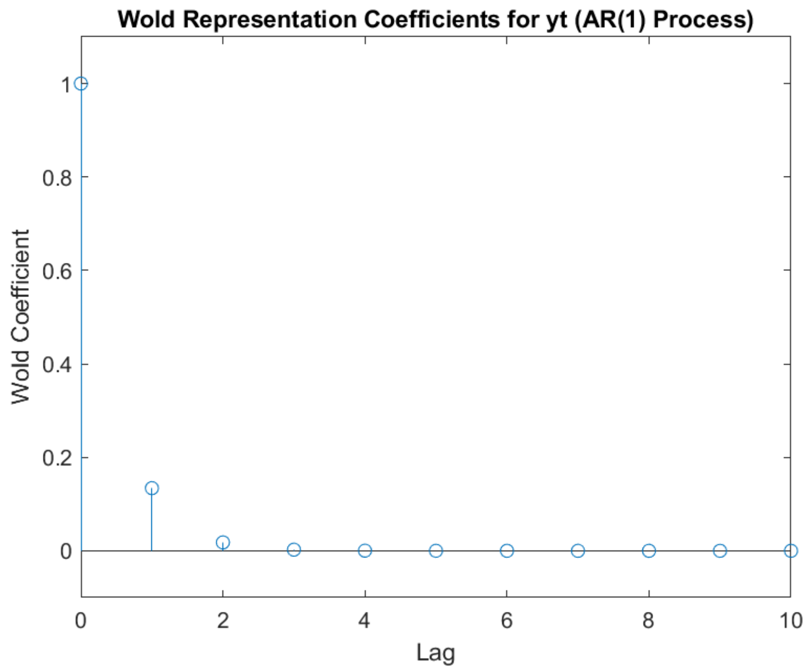
$$y_t = (1 + \phi)c + \phi^2 y_{t-2} + \phi \varepsilon_{t-1} + \varepsilon_t \quad (8)$$

Now, substituting y_{t-2} with its expression and doing this for every $k \leqslant t$, we obtain:

$$y_t = c \sum_{k=0}^{\infty} \phi^k + \phi^k y_{t-k} + \sum_{k=0}^{\infty} \phi^k \varepsilon_{t-k} \quad (9)$$

So the coefficients of the Wold Decomposition are given by ϕ^k.

```matlab
% Define the AR(1) process parameters (replace with
   your estimated values)
% phi: Autoregressive coefficient (estimated value)

% Define the maximum lag for the coefficients
maxLag = 10;

% Create a unit impulse at lag 0
impulse = [1; zeros(maxLag, 1)];

% Use the filter function to obtain the Wold
   coefficients
WoldCoefficients = filter(1, [1, -phi], impulse);

% Plot the Wold coefficients
figure;
stem(0:maxLag, WoldCoefficients);
xlabel('Lag');
ylabel('Wold Coefficient');
ylim([min(WoldCoefficients) - 0.1, max(
   WoldCoefficients) + 0.1]);
title('Wold Representation Coefficients for yt (AR(1)
   Process)');

% Display the Wold coefficients
fprintf('Wold Coefficients for AR(1) Process:\n');
for lag = 0:maxLag
    fprintf('Lag %d: %f\n', lag, WoldCoefficients(lag
       + 1));
end
```


Wold Representation Coefficients for yt (AR(1) Process)
Wold Coefficient
1
0.8
0.6
0.4
0.2
0
0
2
4
6
8
10
Lag

Exercise 2:
Now suppose that GDP admits an AR(2) representation:

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \varepsilon_t$$

(a) Estimate the parameters c, ϕ_1, and ϕ_2 with OLS.

Using the previously defined function ARestimation(p, data), we are able to compute the estimation of both c and the vector of ϕ:

```
[c,phi] = ar_estimation(2,gdp_growth);
fprintf('Estimated Mean: %f\n', c);
fprintf('Estimated Autoregressive Coefficient (phi1):
   %f\n', phi(1));
fprintf('Estimated Autoregressive Coefficient (phi2):
   %f\n', phi(2));

Estimated Mean: 0.609010
Estimated Autoregressive Coefficient (phi1): 0.118662
Estimated Autoregressive Coefficient (phi2): 0.098286
```

So, we can write our estimated AR(2) as:

$$y_t = \hat{c} + \hat{\phi}_1 y_{t-1} + \hat{\phi}_2 y_{t-2} + \varepsilon_t \quad (11)$$

$$y_t = 0.6090 + 0.1187 y_{t-1} + 0.0983 y_{t-2} + \varepsilon_t \quad (12)$$

(b) Compute the roots of the AR polynomial.

In order to find the roots of the AR component of y_t, we have to rewrite our process using a lag polynomial:

$$y_t = (0.6090 + 0.1189L + 0.0983L^2) y_t + \varepsilon_t \quad (13)$$

$$\alpha(L) = (0.6090 + 0.1189L + 0.0983L^2) \quad (14)$$

Now, we set the polynomial equal to zero:

$$\alpha(x) = (0.6090{+}0.1189x{+}0.0983x^2) = 0 \implies |x_1| = 3.17 \text{ and } |x_2| = 1.96 \quad (15)$$

the matlab code is the following

```
%see that the polynomial is given by \phi(z) = 1 - 1.2
    z - 0.10L^2 = 0 we can
%use the root function of matlab to compute the roots
    of that polynomial,

% Define the coefficients a, b, and c
a = -phi(2,1);
b = -phi(1,1);
c = 1;

% Solve the quadratic equation
roots_1 = roots([a, b, c]);
abs_roots = abs(roots_1);

abs_roots % since abs_roots are outside the unit
    circle, the process is stationary
```

(c) State the conditions under which the process is causal and stationary.

Computing the absolute value of the two roots, we can see that both are greater than 1: $|\beta_1| > 1$ and $|\beta_2| > 1$.
This implies that the AR(2) process is Stationary and Causal.

(d) Obtain and plot the coefficients of the Wold representation of y_t.

Wold's decomposition theorem states that it is possible to represent the zero-mean process $\hat{y}_t = y_t - c$ as an infinite sum:

$$\hat{y}_t = \sum_{j=0}^{\infty} \theta_j \varepsilon_{t-j} + \eta_t$$

where $\sum_{j \leqslant 0} \theta_j^2 < \infty$, $\theta_0 = 1$, and the sequence of η_t is linearly deterministic. In our case, since the process is causal, Wold's decomposition coefficients coincide with the ones of the MA(∞) representation of $\hat{y}_t$. Letting $\Phi(L)\hat{y}_t = \varepsilon_t$ implies that:

$$\hat{y}_t = \frac{1}{1 - \phi_1 L - \phi_2 L^2} \varepsilon_t$$

Our problem to solve is now to find the multiplicative inverse of the polynomial $\Phi(z) = 1 - \phi_1 z - \phi_2 z^2$. It is easy to see that:

$$1 - \phi_1 z - \phi_2 z^2 = (1 - \frac{z}{\beta_1})(1 - \frac{z}{\beta_2})$$

Then we can find the constants C_1 and C_2 such that:

$$\frac{1}{(1 - \frac{z}{\beta_1})(1 - \frac{z}{\beta_2})} = \frac{C_1}{1 - \frac{z}{\beta_1}} + \frac{C_2}{1 - \frac{z}{\beta_2}}$$

The constants can be easily obtained, and then we can exploit the geometric series to obtain the coefficients of the Wold representation. Indeed:

$$C_1 = \frac{\beta_2}{\beta_2 - \beta_1}, \quad C_2 = \frac{\beta_1}{\beta_2 - \beta_1}$$

Hence, omitting some computations:

$$\frac{1}{1 - \phi_1 z - \phi_2 z^2} = \frac{1}{\beta_2 - \beta_1} \sum_{j=0}^{\infty} \left(\frac{\beta_2}{\beta_1^j} - \frac{\beta_1}{\beta_2^j} \right) z^j$$

This implies that:

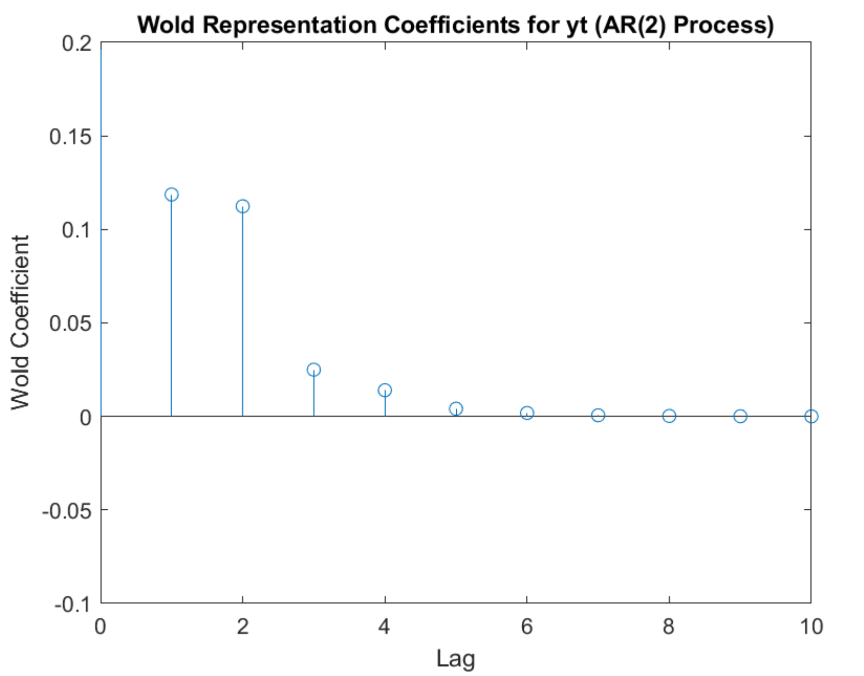
$$\hat{y}_t = \frac{1}{\beta_2 - \beta_1} \sum_{j=0}^{\infty} \left(\frac{\beta_2}{\beta_1^j} - \frac{\beta_1}{\beta_2^j} \right) \epsilon_{t-j}$$

And, therefore, the j-th coefficient of the Wold representation of y_t, for all $j \in \mathbb{N}$, is:

$$\theta_j = \frac{1}{\beta_2 - \beta_1} \left(\frac{\beta_2}{\beta_1^j} - \frac{\beta_1}{\beta_2^j} \right)$$

```
% Define the AR(2) process parameters (replace with
   your estimated values)

% Define the maximum lag for the coefficients
maxLag = 10;

% Create a unit impulse at lag 0
impulse = [1; zeros(maxLag, 1)];

% Use the filter function to obtain the Wold
   coefficients for the AR(2) process
AR2Coefficients = [1, -phi(1), -phi(2)];
WoldCoefficients = filter(1, AR2Coefficients, impulse)
   ;
```

```
% Plot the Wold coefficients
figure;
stem(0:maxLag, WoldCoefficients);
xlabel('Lag');
ylabel('Wold Coefficient');
ylim([min(-0.1), max(0.2)]);
title('Wold Representation Coefficients for yt (AR(2)
    Process)');

% Print the Wold coefficients
fprintf('Wold Representation Coefficients for AR(2)
    Process:\n');
for lag = 0:maxLag
    fprintf('Lag %d: %f\n', lag, WoldCoefficients(lag
        + 1));
end
```



Exercise 3:

Consider the following MA

$$y_t = c + \varepsilon_t + 1.2\varepsilon_{t-1} + 2\varepsilon_{t-2} \tag{1}$$

a) Show that the process is stationary

To demonstrate stationarity, we need to verify the following conditions:

$$E[y_t] = \mu \quad \text{for all } t$$

$$\text{var}[y_t] = E[y_t]^2 < \infty \quad \text{for all } t$$

$$\text{Cov}[y_t, y_{t+\tau}] = \text{Cov}[y_{t+\phi}, y_{t+\tau+\phi}] \quad \text{for all } t, \tau, \phi$$

In summary, weak stationarity means that the mean and variance are finite and constant, and the autocovariance function only depends on ϕ, the distance between observations.

First, we will show that $E[y_t] = \mu$;

$$E[y_t] = E[c + \varepsilon_t + 1.2\varepsilon_{t-1} + 2\varepsilon_{t-2}]$$

$$E[y_t] = E[c] + E[\varepsilon_t] + 1.2E[\varepsilon_{t-1}] + 2E[\varepsilon_{t-2}]$$

Recalling that by assumption, $E[\varepsilon_t] = 0$ for all t. Thus, it's clear that $E[\varepsilon_t] = 1.2E[\varepsilon_{t-1}] = 2E[\varepsilon_{t-2}] = 0$. Therefore, we have simply that:

$$E[y_t] = E[c] = c = \mu < \infty \quad \text{for all } t$$

As desired.

Secondly, we will show that $var[y_t] = E[y_t]^2 < \infty$ for all t. For this, let $\theta_1 = 1.2$ and $\theta_2 = 2$.

Notice that:

$$var[y_t] = var[c] + var[\varepsilon] + \theta_1 var[\varepsilon_{t-1}] + \theta_2 var[\varepsilon_{t-2}]$$

Since c is a constant, it must be that $var[c] = 0$. Then, we have:

$$var[y_t] = var[\varepsilon] + \theta_1 var[\varepsilon_{t-1}] + \theta_2 var[\varepsilon_{t-2}]$$

$$var[y_t] = \sigma_\varepsilon^2 + \theta_1\sigma_\varepsilon^2 + \theta_2\sigma_\varepsilon^2 = \sigma_\varepsilon^2(1+\theta_1+\theta_2) < \infty \quad \text{for all } t$$

As desired.

Finally, we will show that $Cov[y_t, y_{t+\tau}] = Cov[y_{t+\phi}, y_{t+\tau+\phi}]$ for all t, τ, ϕ. For this, again, let $\theta_1 = 1.2$ and $\theta_2 = 2$. In this case, we have:

$$Cov[y_t, y_{t+\tau}] = E[(y_t - \mu)(y_{t+\tau} - \mu)]$$

Then, for $\tau = 0$:

$$Cov[y_t, y_{t+\tau}] = Cov[y_t, y_t] = var[y_t] = \sigma_\varepsilon^2(1+\theta_1+\theta_2)$$

That does not depend on time.

For $\tau = -1$:

$$Cov[y_t, y_{t-1}] = E\left[(c + \varepsilon_t + \theta_1\varepsilon_{t-1} + \theta_2\varepsilon_{t-2} - \mu)\left(c + \varepsilon_{t-1} + \theta_1\varepsilon_{t-2} + \theta_2\varepsilon_{t-3} - \mu\right)\right]$$

And so on for $\tau = +1$ and $\tau = \pm 2$. For $\tau > \pm 2$, $Cov[y_t, y_{t+\tau}] = 0$, which does not depend on time, proving that $Cov[y_t, y_{t+\tau}]$ does not depend on time but only on the distance τ. Thus, the process y_t is (weakly) stationary.

(b) Find the roots of the MA polynomial. Is this process invertible?

Recall that the process is:

$$y_t = c + \varepsilon_t + 1.2\varepsilon_{t-1} + 2\varepsilon_{t-2}$$

Also, recall that a MA(q) process defined by the equation $y_t = \theta(L)\varepsilon_t$ is said to be invertible if and only if $\theta(z) \neq 0$ for all $z \in \mathbb{C}$ such that $\|z\| < 1$.

We consider $\hat{y}_t = y_t - c$, which represents deviations from the unconditional mean of the process:

$$\hat{y}_t = \varepsilon_t + 1.2\varepsilon_{t-1} + 2\varepsilon_{t-2}$$

The relevant polynomial of this process is given by:

$$\hat{y}_t = \underbrace{(1L^0 + 1.2L^1 + 2L^2)}_{\theta(L)}\varepsilon_t$$

$$\theta(L) = 1L^0 + 1.2L^1 + 2L^2$$

$$\theta(L) = 1 + 1.2L + 2L^2$$

We have to check for the solutions of the characteristic equation $\theta(z) = 0$, which is:

$$\theta(z) = 1 + 1.2z + 2z^2 = 0$$

So, the roots of the MA polynomial are given by:

$$z_1 = -0.3 + i6.40 \quad \text{and} \quad z_2 = -0.3 - i6.40$$

Are these roots inside or outside the unit circle? The modulus of a complex number is given by $\|x + iy\| \equiv \sqrt{x^2 + y^2}$:

$$\|z_1\| = \sqrt{(-0.3)^2 + (0.640)^2} = 0.71 < 1$$

$$\|z_2\| = \sqrt{(-0.3)^2 + (0.640)^2} = 0.71 < 1$$

Since the roots are inside the unit circle, the process is not invertible. We have found $z_1, z_2 \in \mathbb{C}$ such that $\|z_1, z_2\| < 1$ for which $\theta(z) = 0$. Moreover, the MA(2) process cannot be expressed as an infinite order AR model that converges.