Time Series: A First Course with Bootstrap Starter

Contents

Lesson 5-1: ARMA Processes

- ARMA processes generalize the AR and MA processes, and are central to classical time series analysis. They are very useful for modeling and forecasting stationary time series data.

Definition 5.1.1.

- $\{X_t\}$ is an ARMA(p,q) process if it is stationary and satisfies

$$X_t - \sum_{j=1}^{p} \phi_j \, X_{t-j} = Z_t + \sum_{j=1}^{q} \theta_j \, Z_{t-j},$$

  where $Z_t \sim \mathrm{WN}(0, \sigma^2)$. The $\{Z_t\}$ process is called the *inputs*.
- This is a recursive definition. It requires p initial conditions to start the process.
- An ARMA is like an AR process with MA inputs.
- Special cases: $p = 0$ gives an MA(q), and $q = 0$ gives an AR(p).

Paradigm 5.1.3. ARMA as a Linear Filter

- We can compactly write the ARMA equation in terms of the backward shift operator B. Define the polynomials

$$\phi(z) = 1 - \sum_{j=1}^{p} \phi_j z^j \qquad \theta(z) = 1 + \sum_{j=1}^{q} \theta_j z^j.$$

  Then the ARMA process satisfies

$$\phi(B) X_t = \theta(B) Z_t.$$

- Note that $\phi_0 = 1$ and $\theta_0 = 1$ in these polynomials.

Example 5.1.4. MA(q) Autocovariance

- Take $p = 0$ but $q > 0$, and determine the autocovariance function.
- Suppose $h \geq 0$:

$$\mathbf{E}[X_t X_{t+h}] = \sum_{j=0}^{q} \sum_{k=0}^{q} \theta_j \theta_k \mathbb{E}[Z_{t-j} Z_{t+h-k}] = \sum_{j=0}^{q-h} \theta_j \theta_{j+h} \, \sigma^2,$$

  where the sum is interpreted as zero if $q < h$.
- The second equality follows from white noise: $\mathbb{E}[Z_{t-j} Z_{t+h-k}] = 0$ unless $t - j = t + h - k$, or $j = k - h$. So the double sum collapses to a single sum, setting $k = j + h$.
- But if $h > q$, then it's impossible for $k = j + h$, because $j + h \geq h > q \geq k$.
- This formula is a convolution of the sequence $\{\theta_j\}$ with its reverse!

Q here : fully get this convolution stuff

Exercise 5.14. Product of Polynomials

- We can quickly compute the autocovariance function for an MA process by convolution of the moving average polynomial with its reverse.
- The convolution is also obtained by reading off the product of polynomials.

. . . by reading off

- We apply this idea to numerically compute the autocovariance for an MA(3) with $\theta(z) = 1 + .4z + .2z^2 - .3z^3$, and $\sigma = 1$.

- First we write a routine to multiply polynomials.

```
polymul <- function(a,b)
{
    bb <- c(b,rep(0,length(a)-1))
    B <- toeplitz(bb)
    B[lower.tri(B)] <- 0
    aa <- rev(c(a,rep(0,length(b)-1)))
    prod <- B %*% matrix(aa,length(aa),1)
    return(rev(prod[,1]))
}
```

- Then we define the polynomial, and take its product with itself *reversed*.
- This will yield the autocovariance at lags $-3, -2, -1, 0, 1, 2, 3$.

```
theta <- c(1,.4,.2,-.3)
gamma <- polymul(theta,rev(theta)) # les coeffs et les coeffs à l'envers
print(gamma)
```

```
## [1] -0.30  0.08  0.42  1.29  0.42  0.08 -0.30
```

Example 5.1.5. MA(∞) Process.

- Letting $q = \infty$ in Example 5.1.4, we obtain the important MA(∞) process: $X_t = \sum_{j \geq 0} \theta_j Z_{t-j}$.
- Assumes the coefficients satisfy $\sum_{j \geq 0} \theta_j^2 < \infty$.
- The autocovariance function is

$$\gamma(h) = \sigma^2 \sum_{j=0}^{\infty} \theta_j \theta_{j+h}.$$

Lesson 5-2: Difference Equations

- To understand ARMA processes it is useful to study *difference* equations, which are a discrete analogue of differential equations.

Definition 5.2.2.

- The equation $\phi(B) X_t = W_t$ is a *linear ordinary difference equation* (ODE) for $\{X_t\}$ with input $\{W_t\}$.
- If $\phi(z)$ has degree p, then the ODE has order p.
- If $W_t \equiv 0$, the ODE is *homogeneous*.

Paradigm 5.2.7. Solution of a Homogeneous ODE.

- The key is to obtain the roots of $\phi(z)$.
- Why: let ζ be a root, so that $\phi(\zeta) = 0$. Then check that $X_t = \zeta^{-t}$ solves the homogeneous ODE:

$$\phi(B) X_t = \sum_{j=0}^{p} \phi_j X_{t-j} = \sum_{j=0}^{p} \phi_j \zeta^{-t+j} = \zeta^{-t} \sum_{j=0}^{p} \phi_j \zeta^j = \zeta^{-t} \phi(\zeta) = 0.$$

- The polynomial has p roots $\zeta_1, \ldots, \zeta_p$. These can be complex numbers, and might be repeated or distinct.
- If the roots are distinct, the general solution has the format

$$X_t = \sum_{j=1}^{p} b_j \zeta_j^{-t},$$

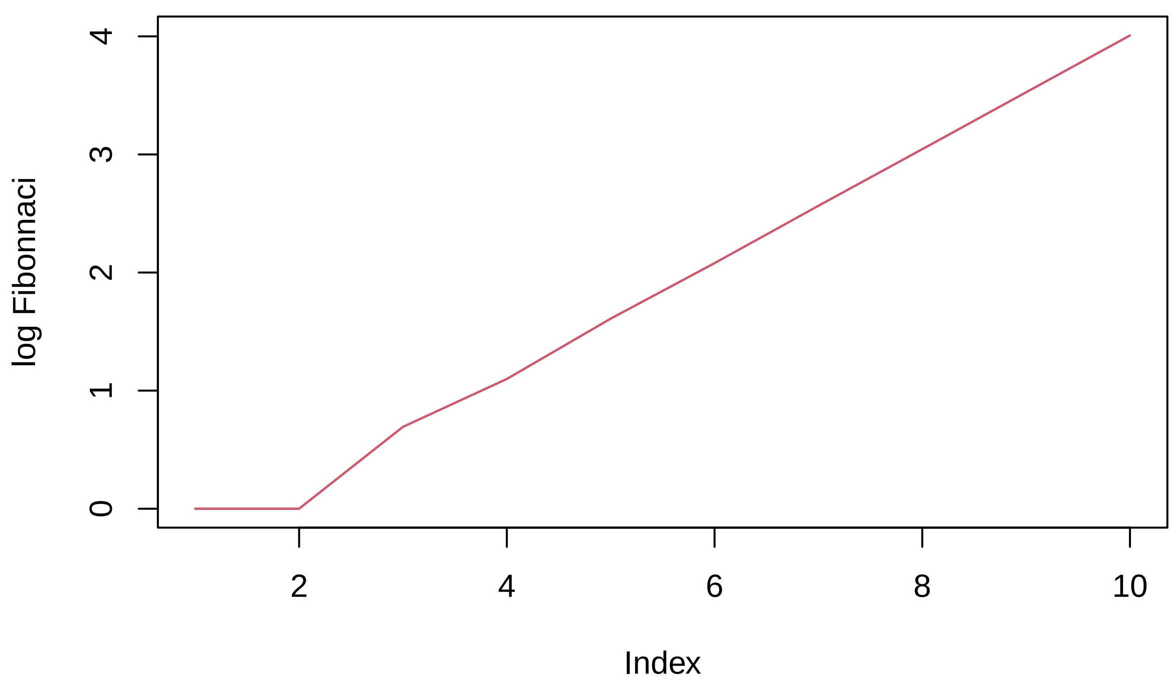
where b_j are coefficients to be determined by the initial conditions.

- Initial conditions are specified values for $X_1, \ldots, X_p$. If these are real, then the solution X_t for $t > p$ will also be real. (Though the coefficients b_j might be complex.)

Example 5.2.10. Fibonacci Sequence.

- Consider the Fibonacci recursion $X_t = X_{t-1} + X_{t-2}$, which corresponds to an ODE with $\phi(z) = 1 - z - z^2$.
- The roots are distinct and real: $\zeta_1 = -1/2 + \sqrt{5}/2$, $\zeta_2 = -1/2 - \sqrt{5}/2$.
- With initial conditions $X_1 = X_0 = 1$, the coefficients are found to be $b_1 = -\zeta_2/\sqrt{5}$ and $b_2 = \zeta_1/\sqrt{5}$.
- Then $X_t = b_1 \zeta_1^{-t} + b_2 \zeta_2^{-t}$ is the homogeneous solution.
- We plot the sequence (in log scale) with this initialization.

```
n <- 10
x <- rep(1,n)
for(i in 3:n) { x[i] <- x[i-1] + x[i-2] }
zeta1 <- (-1 + 5^{1/2})/2
zeta2 <- (-1 - 5^{1/2})/2
b1 <- -zeta2/5^{1/2}
b2 <- zeta1/5^{1/2}
y <- b1*zeta1^{-seq(0,n-1)} + b2*zeta2^{-seq(0,n-1)}
plot(ts(log(x)),xlab="Index",ylab="log Fibonnaci")
lines(ts(log(y)),col=2)
```



Example 5.2.11. Seasonal Difference.

- Any periodic function can be written as a sum of cosines and sines. Why?
- If X_t is periodic with integer period s, then it is annihilated by seasonal differencing, and $(1 - B^s)X_t = 0$.
- So $\phi(z) = 1 - z^s$, which has roots $\zeta_j = e^{2\pi i j/s}$ for $j = 1, \ldots, s$.

- Thus the solution is

$$X_t = \sum_{j=1}^{s} b_j e^{-2\pi i j t/s}.$$

- If s is even, then two roots (corresponding to $j = s/2, s$) are real, and the rest are complex conjugate pairs.
- If s is odd, then one root (corresponding to $j = s$) is real, and the rest are complex conjugate pairs.
- The sequence X_t is real, so $b_j e^{-2\pi i j t/s}$ must be real, and hence

$$X_t = \sum_{j=1}^{s} \mathcal{R}[b_j] \cos(2\pi j t/s) + \mathcal{I}[b_j] \sin(2\pi j t/s).$$

Lesson 5-3: Causality of AR(1)

- Causality is the concept that the present value of a time series does not depend on future values, only on present and past values.

Paradigm 5.3.3. The Causal AR(1) Case.

- From the AR(1) recursion with $|\phi_1| < 1$,

$$X_t = \phi_1 \, X_{t-1} + Z_t,$$

  and we can recursively solve.
- So we obtain

$$X_t = \phi_1 \, (\phi_1 X_{t-2} + Z_{t-1}) + Z_t = \phi_1^2 \, X_{t-2} + \phi_1 \, Z_{t-1} + Z_t.$$

- Iterating this argument further, we obtain

$$X_t = \phi_1^t \, X_0 + \phi_1^{t-1} \, Z_1 + \ldots + \phi_1 \, Z_{t-1} + Z_t.$$

- Going further into the past, we obtain

$$X_t = Z_t + \phi \, Z_{t-1} + \ldots = \sum_{j \geq 0} \phi_1^j \, Z_{t-j}.$$

- So X_t only depends on present and past variables $\{Z_t\}$. This gives a causal representation.

Remark 5.3.4.

- The ODE $(1 - \phi_1 B) X_t = Z_t$ is solved by

$$X_t = \sum_{j \geq 0} \phi_1^j \, Z_{t-j}.$$

- Check:

$$X_t = Z_t + \sum_{j \geq 1} \phi_1^j Z_{t-j} = Z_t + \sum_{j \geq 0} \phi_1^{j+1} Z_{t-j-1} = Z_t + \phi_1 \sum_{j \geq 0} \phi_1^j Z_{t-1-j} = Z_t + \phi_1 \, X_{t-1}.$$

Example 5.3.7. Causal AR(1) Autocovariance.

- We see that the causal AR(1) solution corresponds to an MA(∞) with $\theta_j = \phi_1^j$.
- Therefore the autocovariance for $h \geq 0$ is given by

$$\gamma(h) = \sigma^2 \sum_{j=0}^{\infty} \phi_1^j \phi_1^{j+h} = \sigma^2 \, \phi_1^h \sum_{j=0}^{\infty} \phi_1^{2j} = \sigma^2 \, \phi_1^h / (1 - \phi_1^2).$$

- So the variance is $\sigma^2/(1 - \phi_1^2)$, and $\rho(h) = \phi_1^{|h|}$.

Lesson 5-4: Causality of ARMA

- Causality is a useful concept for forecasting, and can be used to derive the h-step ahead forecast filter.

Definition 5.4.1.

- The ARMA process $\{X_t\}$ is *causal* with respect to its inputs $\{Z_t\}$ if there exists a power series $\psi(z) = \sum_{j\geq 0} \psi_j z^j$ such that
$$X_t = \psi(B) Z_t = \sum_{j\geq 0} \psi_j \, Z_{t-j}.$$
- This is called the MA(∞) representation, since it expresses $\{X_t\}$ as an MA(∞) process.

Theorem 5.4.3.

- Let $\{X_t\}$ be an ARMA(p,q) where $\phi(z)$ and $\theta(z)$ have no common roots. Then $\{X_t\}$ is causal if and only if all the roots of $\phi(z)$ are outside the unit circle, i.e., $|z| > 1$ when $\phi(z) = 0$. In this case,
$$\psi(z) = \sum_{j\geq 0} \psi_j z^j = \frac{\theta(z)}{\phi(z)}.$$
- The coefficients ψ_j can be computed by recursions, by partial fraction decomposition, or by the theory of ODE.

Remark 5.4.5. Common Roots.

- If the AR and MA polynomials had a common root, it could be cancelled from both polynomials, yielding a simplified difference equation.
- For example: $X_t - .5X_{t-1} = Z_t - .5Z_{t-1}$ has the solution $X_t = Z_t$, given by cancellation.

Exercise 5.26. Cancellation in an ARMA(1,2).

- Suppose that $X_t - .5X_{t-1} = Z_t - 1.3Z_{t-1} + .4Z_{t-2}$.
- This is equivalent to an MA(1) process: $\phi(z) = 1 - .5z$, and
$$\theta(z) = 1 - 1.3z + .4z^2 = (1 - .5z)(1 - .8z).$$
- Thus $X_t = Z_t - .8Z_{t-1}$.

Lesson 5-5: Invertibility of ARMA

- Some processes also have an infinite order *autoregressive* representation.

Definition 5.5.1.

- The ARMA process $\{X_t\}$ is *invertible* with respect to its inputs $\{Z_t\}$ if there exists a power series $\pi(z) = \sum_{j\geq 0} \pi_j z^j$ such that
$$Z_t = \pi(B) X_t = \sum_{j\geq 0} \pi_j \, X_{t-j}.$$
- This is called the AR(∞) representation, since it represents $\{X_t\}$ as an autoregressive process of infinite order.
- Invertibility is crucial for prediction applications, because it guarantees the non-singularity of certain covariance matrices needed for prediction.

Theorem 5.5.3.

- Let $\{X_t\}$ be an ARMA(p,q) where $\phi(z)$ and $\theta(z)$ have no common roots. Then $\{X_t\}$ is invertible if and only if all the roots of $\theta(z)$ are outside the unit circle, i.e., $|z| > 1$ when $\theta(z) = 0$. In this case,

$$\pi(z) = \sum_{j \geq 0} \pi_j z^j = \frac{\phi(z)}{\theta(z)}.$$

- The coefficients π_j can be computed by recursions, by partial fraction decomposition, or by the theory of ODE.

Example 5.5.7. ARMA(1,2) Process

- Consider the ARMA(1,2) process

$$X_t - (1/2)X_{t-1} = Z_t + (5/6)Z_{t-1} + (1/6)Z_{t-2}.$$

- So $\phi(z) = 1 - (1/2)z$ and $\theta(z) = (1 + (1/2)z)(1 + (1/3)z)$.
- Since the root of $\phi(z)$ is $z = 2$, which has magnitude larger than one, the process is causal. Then

$$\phi(z)\psi(z) = \theta(z).$$

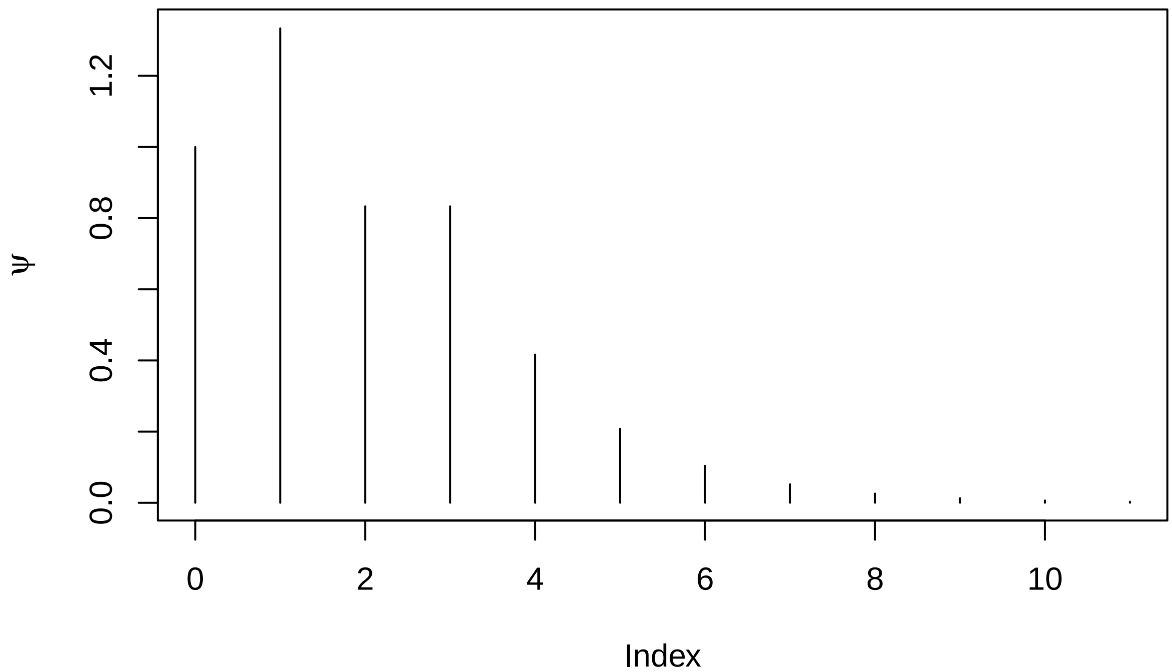
  By matching coefficients,

$$\psi_k - (1/2)\psi_{k-1} = \theta_k$$

  for $k \geq 0$, where $\psi_k = 0$ if $k < 0$. Also $\theta_k = 0$ if $k > 2$, while $\theta_0 = 1$, $\theta_1 = 5/6$, and $\theta_2 = 1/6$. Solving recursively, we get

$$\begin{aligned}
\psi_0 &= 1 \\
\psi_1 &= 4/3 \\
\psi_2 &= 5/6 \\
\psi_k &= (1/2)\psi_{k-1} \qquad k \geq 3.
\end{aligned}$$

```r
psi <- c(1,4/3,5/6,(10/3)*(1/2)^seq(2,10))
plot(ts(psi,start=0),type="h",xlab="Index",ylab=expression(psi))
```



- Since the roots of $\theta(z)$ are $z = -2, -3$, which have magnitude larger than one, the process is invertible. Then

  $$\theta(z)\pi(z) = \phi(z).$$

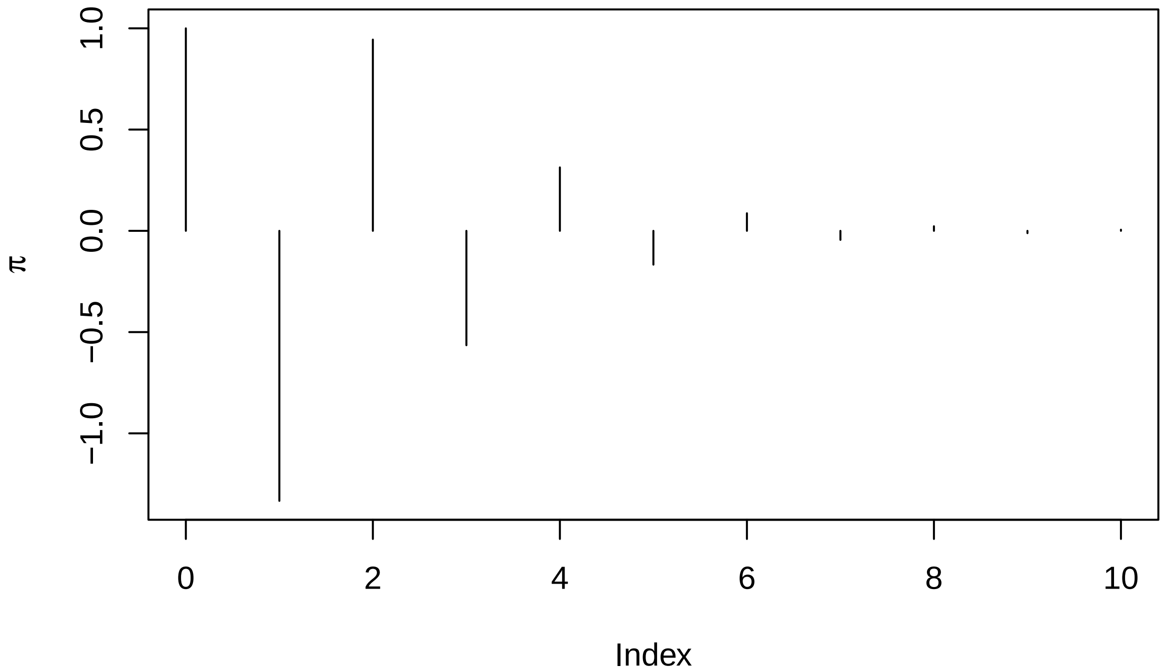
  By matching coefficients,

  $$\pi_k + (5/6)\pi_{k-1} + (1/6)\pi_{k-2} = \begin{cases} 1 & \text{if } k = 0 \\ -1/2 & \text{if } k = 1 \\ 0 & \text{if } k \geq 2 \end{cases}$$

  for $k \geq 0$, where $\pi_k = 0$ if $k < 0$. Solving recursively, we get

  $$\begin{aligned} \pi_0 &= 1 \\ \pi_1 &= -4/3 \\ \pi_k &= -(5/6)\pi_{k-1} - (1/6)\pi_{k-2} \qquad k \geq 2. \end{aligned}$$

```
pi <- c(1,-4/3)
for(j in 2:10)
{
  pi <- c(pi,(-5/6)*pi[j]+(-1/6)*pi[j-1])
}
plot(ts(pi,start=0),type="h",xlab="Index",ylab=expression(pi))
```



Lesson 5-6: Autocovariance Generating Function

- We want a way to summarize the autocovariances for an ARMA process.

Definition 5.6.1.

The *autocovariance generating function* (AGF) of a stationary time series with autocovariance function $\gamma(k)$ is

$$G(z) = \sum_{k=-\infty}^{\infty} \gamma(k) z^k$$

(if it converges in some annulus $1/r < |z| < r$ for $r > 1$).

Example 5.6.3. Constant AGF

- Suppose $X_t \sim \mathrm{WN}(0, \sigma^2)$. Then $\gamma(0) = \sigma^2$, and $\gamma(k) = 0$ if $k \neq 0$. Hence

$$G(z) = \gamma(0) = \sigma^2.$$

- The AGF for white noise is a constant function.

Definition 5.6.5.

- Suppose that $Y_t = \psi(B) X_t$ for some linear filter $\psi(B)$. For complex z, the *transfer function* of the filter is $\psi(z)$. Its coefficients ψ_j are the *impulse response coefficients*.

Theorem 5.6.6.

Suppose we filter stationary $\{X_t\}$ with some $\psi(B)$, yielding $Y_t = \psi(B)X_t$. Then the AGFs of input and output are related by

$$G_y(z) = \psi(z)\psi(z^{-1})G_x(z).$$

Remark 5.6.8. ARMA Transfer Function

- Because a causal ARMA can be written in MA representation as $X_t = \psi(B)Z_t$, we have

  $$G_x(z) = \psi(z)\psi(z^{-1})\sigma^2,$$

  by Example 5.6.3.

- Using $\psi(z) = \theta(z)/\phi(z)$, we obtain

  $$G_x(z) = \frac{\theta(z)\theta(z^{-1})}{\phi(z)\phi(z^{-1})}\sigma^2.$$

Example 5.6.9. MA(1) AGF

- We can use the AGF to compute autocovariances from MA parameters.
- Suppose $\{X_t\}$ is an MA(1) process with polynomial $\theta(z) = 1 + \theta_1 z$. Then

  $$G_x(z) = \theta(z)\theta(z^{-1})\sigma^2 = (1+\theta_1 z)(1+\theta_1 z^{-1})\sigma^2 = \left(1 + \theta_1^2 + \theta_1 z + \theta_1 z^{-1}\right)\sigma^2.$$

- Because the coefficient of $z^0 = 1$ is $\gamma(0)$, we have

  $$\gamma(0) = (1+\theta_1^2)\sigma^2.$$

- Also, the coefficient of both z and z^{-1} is $\gamma(1)$. Therefore

  $$\gamma(1) = \theta_1\sigma^2.$$

Example 5.6.10. AR(1) AGF

- We can also compute the AGF for an AR(1).
- Suppose $\{X_t\}$ is an AR(1) with causal polynomial $\phi(z) = 1 - \phi_1 z$. Then

  $$G_x(z) = \frac{1}{\phi(z)\phi(z^{-1})}\sigma^2 = \frac{1}{(1-\phi_1 z)(1-\phi_1 z^{-1})}\sigma^2.$$

- By geometric series,

  $$(1-\phi_1 z)^{-1} = \sum_{j\geq 0}\phi_1^j z^j.$$

  (Causality guarantees that $|\phi_1| < 1$!)
- Therefore

  $$\begin{aligned} G_x(z) &= \left(\sum_{j\geq 0}\phi_1^j z^j\right)\left(\sum_{j\geq 0}\phi_1^j z^{-j}\right)\sigma^2 \\ &= \sum_{j,k\geq 0}\phi_1^{j+k} z^{j-k}\sigma^2 \\ &= \sum_{h=-\infty}^{\infty}\sum_{k\geq 0}\phi_1^{|h|+2k} z^h \sigma^2 \\ &= \sum_{h=-\infty}^{\infty}\frac{\phi_1^{|h|}}{1-\phi_1^2} z^h\sigma^2. \end{aligned}$$

- Now we read off the coefficient of z^h (or z^{-h}) is $\gamma(h)$:

$$\gamma(h) = \frac{\phi_1^{|h|}}{1 - \phi_1^2} \sigma^2.$$

Lesson 5-7: MA Representation

- We know the autocovariances of an MA process, but how about an ARMA?
- Given the ARMA polynomials $\theta(z)$ and $\phi(z) = 1 - \sum_{j=1}^p \phi_j z^j$, we need algorithms to compute the autocovariances.

Paradigm 5.7.1. Method 1 for ARMA Autocovariances

- First determine the coefficients of $\psi(z)$, the MA representation. Then compute

$$\gamma(h) = \sum_{j \geq 0} \psi_j \psi_{j+|h|} \sigma^2,$$

which follows from the AGF.
- We get ψ_j recursively by using $\psi(z)\phi(z) = \theta(z)$, so that the θ_j coefficients equal the convolution of ψ_j and ϕ_j.
- Letting $\phi_j = 0$ for $j > p$ and $\theta_j = 0$ for $j > q$, we obtain

$$\psi_j = \theta_j + \sum_{k=1}^j \phi_k \psi_{j-k}.$$

- We can also obtain a direct formula using ODE theory.

Example 5.7.2. Cyclic ARMA(2,1) Process

- We define an ARMA(2,1) process with cyclic properties.
- For $\rho \in (0,1)$ and $\omega \in (0, \pi)$, let $\{X_t\}$ satisfy

$$(1 - 2\rho \cos(\omega) B + \rho^2 B^2) X_t = (1 - \rho \cos(\omega) B) Z_t.$$

- The roots of $\phi(z) = 1 - 2\rho\cos(\omega) z + \rho^2 z^2$ are $\rho^{-1} e^{\pm i\omega}$.
- We use ODE theory with initial conditions $\psi_0 = 1$,

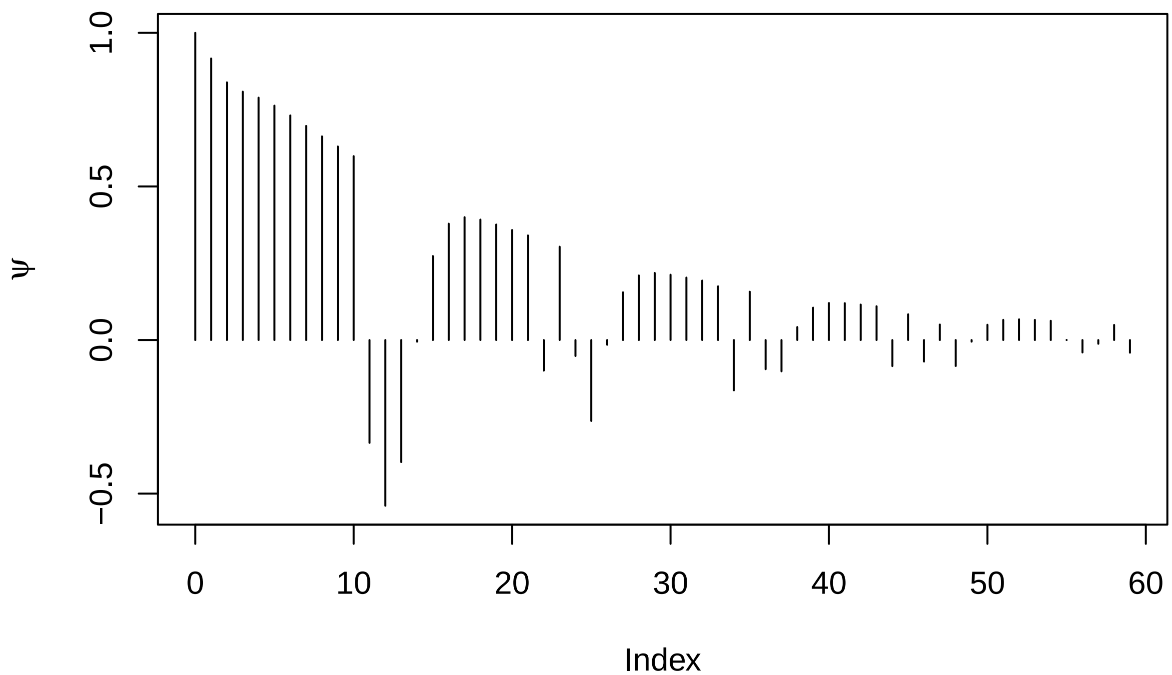
$$\psi_1 = \theta_1 + \psi_0 \phi_1 = \rho \cos(\omega),$$

and eventually find $\psi_j = \rho^j \cos(\omega j)$ for $j \geq 0$.

```
rho <- .95
omega <- pi/5
lag <- 60
psi <- (rho^(seq(1,lag)-1))*cos((seq(1,lag)-1)*omega)
```

```
## Warning in (seq(1, lag) - 1) * omega: la taille d'un objet plus long n'est pas
## multiple de la taille d'un objet plus court
```

```
plot(ts(psi,start=0),type="h",xlab="Index",ylab=expression(psi))
```



- From the MA(∞) representation, we obtain the autocovariance:

$$\gamma(k) = \frac{\sigma^2}{2}\,\rho^k \left(\frac{\cos(\omega k)}{1-\rho^2} + \frac{\cos(\omega k) - \rho^2 \cos(\omega(k-2))}{1 - 2\rho^2\cos(2\omega) + \rho^4} \right).$$

- We rewrite this formula slightly and implement in R.

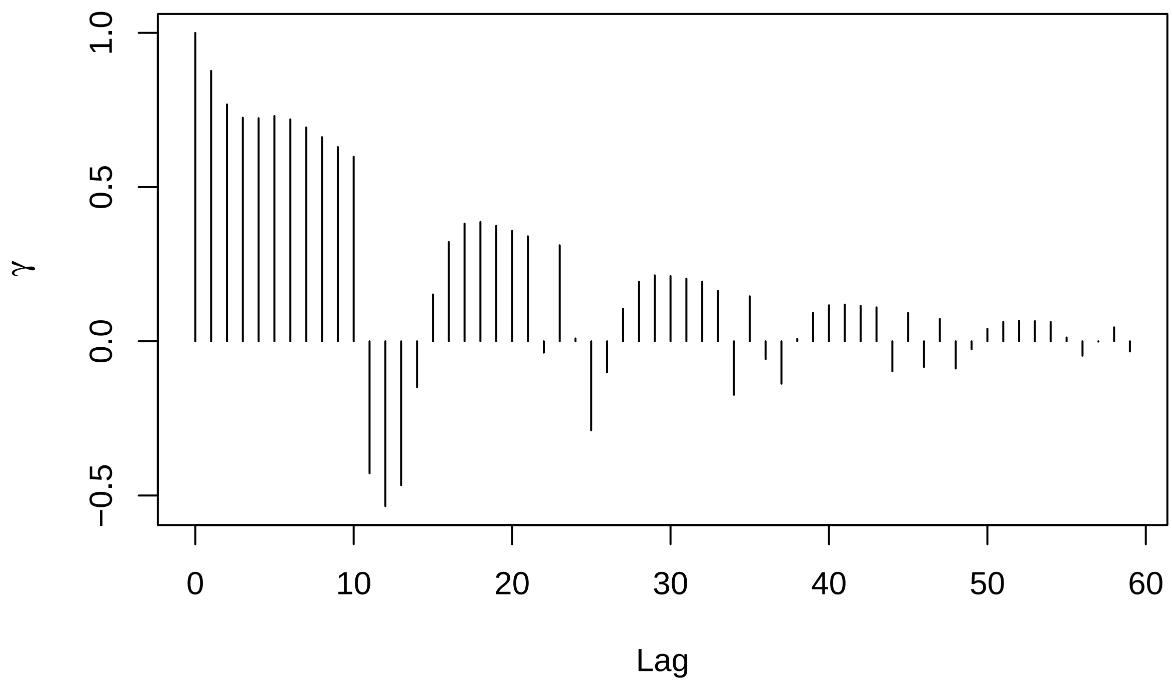
```
const1 <- 1/(1-rho^2) + (1 - rho^2*cos(2*omega))/(1 - 2*rho^2*cos(2*omega) + rho^4)
const2 <- rho^2*sin(2*omega)/(1 - 2*rho^2*cos(2*omega) + rho^4)
gamma <- .5*(rho^(seq(1,lag)-1))*(cos((seq(1,lag)-1)*omega)*const1 - sin((seq(1,lag)-1)*omega)*const2)
```

```
## Warning in (seq(1, lag) - 1) * omega: la taille d'un objet plus long n'est pas
## multiple de la taille d'un objet plus court

## Warning in cos((seq(1, lag) - 1) * omega) * const1: la taille d'un objet plus
## long n'est pas multiple de la taille d'un objet plus court

## Warning in (seq(1, lag) - 1) * omega: la taille d'un objet plus long n'est pas
## multiple de la taille d'un objet plus court

## Warning in sin((seq(1, lag) - 1) * omega) * const2: la taille d'un objet plus
## long n'est pas multiple de la taille d'un objet plus court
```

```
gamma <- gamma/(const1/2)
```

```
## Warning in gamma/(const1/2): la taille d'un objet plus long n'est pas multiple
## de la taille d'un objet plus court
```

```
plot(ts(gamma,start=0),type="h",xlab="Lag",ylab=expression(gamma))
```



Lesson 5-8: Recursive Computation of Autocovariance

- A second technique finds the autocovariances without first finding the MA representation.

Paradigm 5.8.1. Method 2 for ARMA Autocovariances

- Determine a recursive relation for the $\gamma(h)$:

$$\gamma(k) - \sum_{j=1}^{p} \phi_j \gamma(k-j) = \begin{cases} \sigma^2 \sum_{j=0}^{q-k} \theta_{j+k} \psi_j & \text{if } k \leq q \\ 0 & \text{if } k > q. \end{cases}$$

- This is compactly written as $\phi(B)\gamma_k = 0$ for $k > q$, an ODE in terms of the autocovariance function.
- To solve, we find the roots of $\phi(z)$ and determine the homogeneous solution, using initial conditions for γ_k.
- If the roots ζ_j of $\phi(z)$ are distinct, then

$$\gamma(k) = \sum_{j=1}^{p} b_j \zeta_j^{-k}$$

for coefficients b_j.
- These initial conditions can be recursively determined, using other expressions for $\mathbb{E}[W_t X_{t-h}]$, where $W_t = \phi(B) X_t$ (this is the "moving average" portion of the ARMA process).

Proposition 5.8.3. Exponential Decay of ARMA ACF.

Consider a stationary ARMA(p,q) process such that $\phi(B)X_t = \theta(B)Z_t$, for $Z_t \sim \mathrm{WN}(0,\sigma^2)$. Assume ϕ and θ have no common roots. Then there exists a constant $C > 0$ and $r \in (0,1)$ such that

$$|\gamma(k)| \leq C r^{|k|}$$

for all $|k| \geq \max\{p, q+1\}$. Hence the ACF exists.

Exercise 5.51. Direct Algorithm for Autocovariance Function for the ARMA(p,q)

- We encode the second method and run on Example 5.5.7.
- This encoding is *ARMAauto.r*. Most of the code has to do with computing the initial values of the autocovariance, and the latter part of the code has the recursion.

```
polymult <- function(a,b) {
bb <- c(b,rep(0,length(a)-1))
B <- toeplitz(bb)
B[lower.tri(B)] <- 0
aa <- rev(c(a,rep(0,length(b)-1)))
prod <- B %*% matrix(aa,length(aa),1)
return(rev(prod[,1]))
}

ARMAauto <- function(phi,theta,maxlag)
{
    p <- length(phi)
    q <- length(theta)
    gamMA <- polymult(c(1,theta),rev(c(1,theta)))
    gamMA <- gamMA[(q+1):(2*q+1)]
    if (p > 0)
    {
        Amat <- matrix(0,nrow=(p+1),ncol=(2*p+1))
        for(i in 1:(p+1))
        {
            Amat[i,i:(i+p)] <- c(-1*rev(phi),1)
        }
        Amat <- cbind(Amat[,(p+1)],as.matrix(Amat[,(p+2):(2*p+1)]) +
            t(matrix(apply(t(matrix(Amat[,1:p],p+1,p)),2,rev),p,p+1)))
        Bmat <- matrix(0,nrow=(q+1),ncol=(p+q+1))
        for(i in 1:(q+1))
        {
            Bmat[i,i:(i+p)] <- c(-1*rev(phi),1)
        }
        Bmat <- t(matrix(apply(t(Bmat),2,rev),p+q+1,q+1))
        Bmat <- matrix(apply(Bmat,2,rev),q+1,p+q+1)
        Bmat <- Bmat[,1:(q+1)]
        Binv <- solve(Bmat)
        gamMix <- Binv %*% gamMA
        if (p <= q) { gamMix <- matrix(gamMix[1:(p+1),],p+1,1)
            } else gamMix <- matrix(c(gamMix,rep(0,(p-q))),p+1,1)
        gamARMA <- solve(Amat) %*% gamMix
    } else gamARMA <- gamMA[1]

    gamMA <- as.vector(gamMA)
    if (maxlag <= q) gamMA <- gamMA[1:(maxlag+1)] else gamMA <- c(gamMA,rep(0,(maxlag-q)))
```

```
    gamARMA <- as.vector(gamARMA)
    if (maxlag <= p) gamARMA <- gamARMA[1:(maxlag+1)] else {
    for(k in 1:(maxlag-p))
    {
        len <- length(gamARMA)
        acf <- gamMA[p+1+k]
        if (p > 0) acf <- acf + sum(phi*rev(gamARMA[(len-p+1):len]))
        gamARMA <- c(gamARMA,acf)
    } }
    return(gamARMA)
}
```

- We illustrate with particular settings.

```
phi1 <- .5
theta1 <- 5/6
theta2 <- 1/6
sigma <- 1
n <- 10
my.acf <- ARMAauto(phi1,c(theta1,theta2),n)*sigma^2
plot(ts(my.acf,start=0),xlab="Lag",ylab="Autocovariance",
     ylim=c(min(my.acf),max(my.acf)),type="h")
```