Day 19 - ARMA models

Introduction

In this set of notes, we introduce moving average processes, MA(q), which are autoregressive processes on the error terms. We then combine these models with the AR(p) process to develop ARMA(p, q) models, which are valuable for modeling various types of serial autocorrelation in residual error series.

```
# packages
library(tidyverse)
library(lubridate)
```

Moving average processes

Moving average processes

A time series $\{x_t\}$ is a ____ or MA(q) if

$$x_t = w_t + \beta_1 w_{t-1} + \beta_2 w_{t-2} + \cdots + \beta_q w_{t-q}$$

where $\{w_t\}$ is white noise with mean 0 and variance σ_w^2 and the β_i are the model parameters. It can be shown that the MA(q) model can be expressed as a polynomial of order q in the backshift operator:

$$x_t = (1 + \beta_1 B + \beta_2 B^2 + \cdots + \beta_q B^q) w_t = \phi_q(B) w_t$$

A moving average process is said to be _____________________________ if it can be expressed as a stationary autoregressive process of infinite order without an error term. A MA(q) process is invertible if the roots of $\phi_q(B)$ all exceed unity in magnitude.
The autocorrelation function, for $k \geq 0$, is

$$\rho(k) = \begin{cases} 1 & k = 0 \\ \frac{\sum_{i=0}^{q-k} \beta_i \beta_{i+k}}{\sum_{i=0}^{q} \beta_i^2} & k = 1, \ldots, q \\ 0 & k > q \end{cases}$$

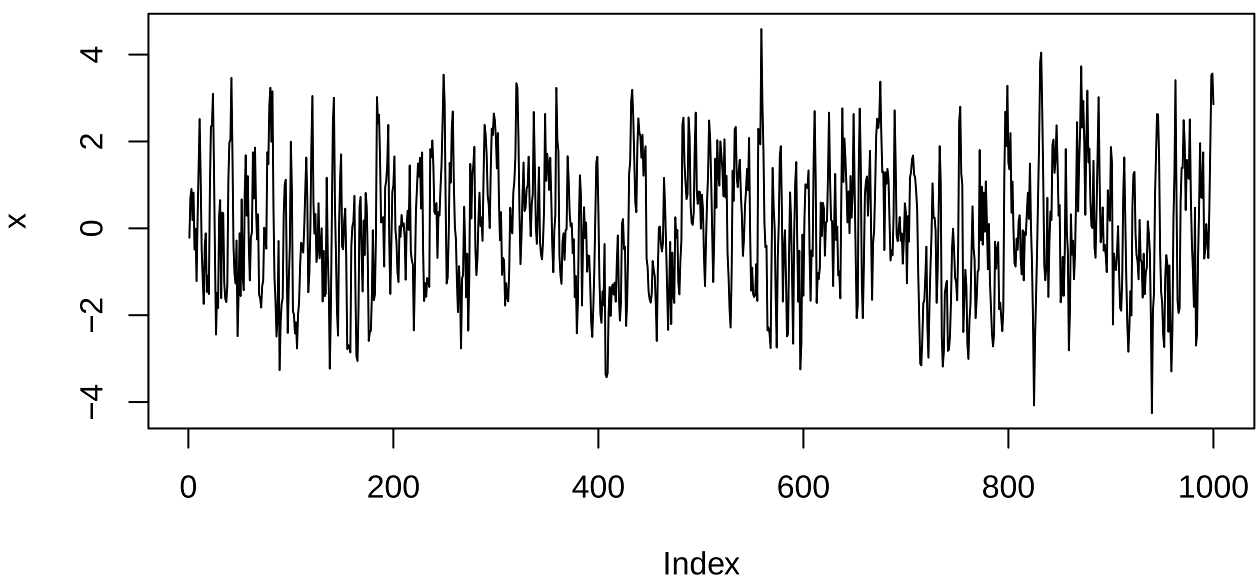
where β_0 is always assumed to be 1.

Express the the MA(2) series: $x_t = w_t + .5w_{t-1} - .4w_{t-2}$ in terms of $\phi_q(B)$ and determine if the process is invertible.

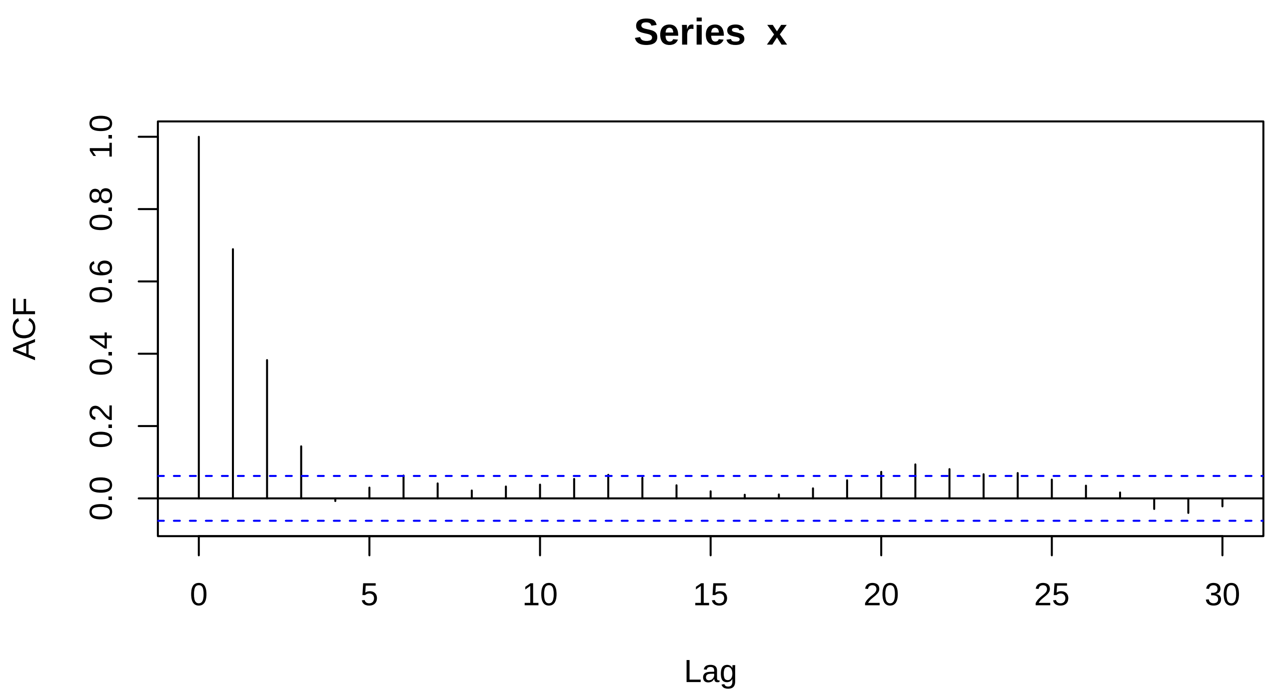
Simulation and fitting

Simulate an MA(3) process with $\beta_1 = .8$, $\beta_2 = .6$, and $\beta_3 = .4$.

```
set.seed(11112024)
x <- w <- rnorm(1000)
b <- c(.8, .6, .4)
for(t in 4:1000){
  for(j in 1:3) x[t] <- x[t] + b[j] * w[t-j]
}
plot(x, type = "l")
```




```
acf(x)
```




```r
ma <- arima(x, order = c(0, 0, 3))
ma
```

```
Call:
arima(x = x, order = c(0, 0, 3))

Coefficients:
         ma1     ma2     ma3  intercept
      0.8195  0.6200  0.3985    -0.0149
s.e.  0.0286  0.0365  0.0313     0.0930

sigma^2 estimated as 1.076:  log likelihood = -1455.87,  aic = 2921.74
```

ARMA(p, q) processes

ARMA processes

A time series $\{x_t\}$ is a ____ or ARMA(p, q) if

$$x_t = \alpha_1 x_{t-1} + \cdots + \alpha_p x_{t-p} + w_t + \beta_1 w_{t-1} + \cdots + \beta_q w_{t-q}$$

where $\{w_t\}$ is white noise with mean 0 and variance σ_w^2. We can express this model using the backshift operator on both x_t and w_t:

$$\begin{aligned} x_t - \alpha_1 x_{t-1} - \cdots - \alpha_p x_{t-p} &= w_t + \beta_1 w_{t-1} + \cdots + \beta_q w_{t-q} \\ (1 - \alpha_1 B - \cdots - \alpha_p B^p) x_t &= (1 + \beta_1 B + \cdots + \beta_q B^q) w_t \\ \theta_p(B) x_t &= \phi_q(B) w_t \end{aligned}$$

The autocorrelation function is reasonably complicated, and I do not expect you to know it.

A few notes about ARMA(p, q) processes:

- The process is stationary if all the roots of $\theta_p(B)$ exceed unity in magnitude.
- The process is invertible if all the roots of $\phi_q(B)$ exceed unity in magnitude.
- Fitting an ARMA(p, q) model will often require less parameters than fitting an AR(p) or MA(q) model on its own. This idea is called *parameter parsimony*.
- When $\theta_p(B)$ and $\phi_q(B)$ share a common factor, a stationary model can be simplified. For example, $(1 - \frac{1}{2}B)(1 - \frac{1}{3}B)x_t = (1 - \frac{1}{2}B)w_t$ can be written as $(1 - \frac{1}{3}B)x_t = w_t$.

Express the following model in ARMA(p, q) notation and determine whether it is stationary and/or invertible. Ensure that the ARMA(p, q) notation is expressed in simplest form.

$$x_t = x_{t-1} - \frac{1}{4}x_{t-2} + w_t + \frac{1}{2}w_{t-1}$$

Simulation and fitting

Complex time series may be simulated using the `arima.sim` function, and fitted using either `arima` or `auto.arima` in the forecast package. The latter using information criterion to select the best stochastic model, ranging from simple AR(p) models to seasonal ARIMA models (more on this next week).

```r
set.seed(11102024)
x <- arima.sim(
  n = 10000,
  model = list(
    ar = c(-.6, .2),
    ma = c(.4, .7)
  )
)
arima(x, order = c(2, 0, 2))
```

```

Call:
arima(x = x, order = c(2, 0, 2))

Coefficients:
          ar1     ar2     ma1     ma2  intercept
      -0.5834  0.2070  0.3770  0.6885    -0.0119
s.e.   0.0131  0.0131  0.0098  0.0085     0.0151

sigma^2 estimated as 1.011:  log likelihood = -14244.75,  aic = 28501.51
```

```r
# auto.arima returns the best ARIMA model using AIC, AICc, or BIC
library(forecast)
auto.arima(x, max.d = 0, max.D = 0, max.P = 0, max.Q = 0)
```

```
Series: x
ARIMA(2,0,2) with zero mean

Coefficients:
          ar1     ar2     ma1     ma2
      -0.5832  0.2071  0.3770  0.6885
s.e.   0.0131  0.0131  0.0098  0.0085

sigma^2 = 1.011:  log likelihood = -14245.06
AIC=28500.13   AICc=28500.14   BIC=28536.18
```

What to know

As the models we consider increase in complexity, it might be helpful to keep track of what is expected of you. You should be able to:

- Write an ARMA(p, q) process in terms of its characteristic polynomials
- Determine whether the ARMA(p, q) process is stationary and/or invertible
- Express the ARMA(p, q) in its simplest form
- Simulate from an ARMA(p, q) process using `arima.sim`
- Fit a particular ARMA(p, q) model using `arima`
- Use `auto.arima` to estimate the best ARMA(p, q) model for an observed data set.
- Notice that the models are fit using `arima` and `auto.arima`, meaning you have access to all the tools introduced with state-space models! You may use `xreg` to specify regression coefficients and `predict` to forecast the series.
- In general, *know when a time series model accounts for the serial autocorrelation that exists within the data.* **No matter what model you fit, the residuals should represent a white-noise series!**