Time Series Analysis

Lecture 3

Autoregressive Models and Moving Average Models

datascience@berkeley

Autoregressive Models,

Model Estimation and Model Selection

Estimation: Example 1–AR(1)

Let's apply an AR model to the series we simulated using an AR(1) process of the following specification.

$$x_t - \mu = 0.7(x_{t-1} - \mu) + \omega_t$$

where μ is the mean of the series

Simulation conducted in R:

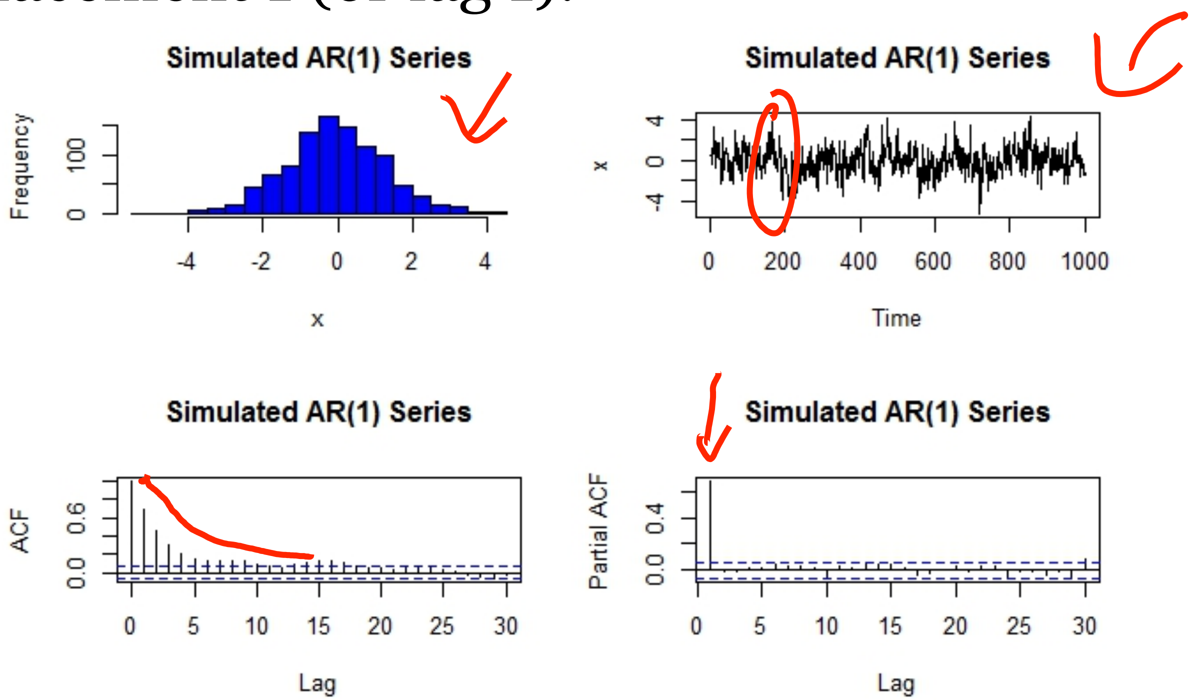
```
x <- w <- rnorm(1000)
x <- arima.sim(n = 1000, list(ar=c(0.7), ma=0))
str(x)
summary(x)
```

Always examine the series after the simulation:

```
> str(x)
 Time-Series [1:1000] from 1 to 1000: 0.5094 0.2424 1.1298 -0.4148 -0.0859 ...
> summary(x)
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-5.30700 -0.92890 -0.11660 -0.07005  0.77120  4.32200
```

Estimation: Example 1–AR(1)

- Examine the simulated series by visualizing its distribution (using histogram), its dynamics (using time series plot), and its dependence structure (using ACF and PACF graphs).
- ACF gradually tapers off to zero; PACF falls off sharply after time displacement 1 (or lag 1).



Estimation: Example 1–AR(1)

- The estimation uses R's `ar()` function.

- This function estimates an autoregressive model of the following form:

  x[t] - m = a[1]*(x[t-1] - m) + … + a[p]*(x[t-p] - m) + e[t]

- Below is the specification of the function in **R.**

```
ar(x, aic = TRUE, order.max = NULL,
   method = c("yule-walker", "burg", "ols", "mle", "yw"),
   na.action, series, ...)
```

- By default it selects the estimated model with the lowest Akaike Information Criterion (AIC; Akaike, 1974).

- AIC is a goodness-of-fit measure that penalizes the number of parameters used in the model. We will discuss AIC next.

Estimation: Example 1–AR(1)

- The commands used to estimate the AR model are listed below. Note that maximum likelihood estimation (MLE) is chosen for the estimation.

```
x.ar <- ar(x, method = "mle")
x.ar$order # order of the AR model with lowest AIC
x.ar$ar # parameter estimate
sqrt(x.ar$asy.var) # asy. standard error
summary(x.ar)
x.ar$aic
```

1. The first command estimates a series of AR models and stores the estimation "object" in `x.ar`; recall that everything in R is an object.
2. The second command displays the order of the AR model that has the lowest AIC.
3. The third command displays the AR parameter estimates associated with this model.

Estimation: Example 1–AR(1)

```
x.ar <- ar(x, method = "mle")
x.ar$order # order of the AR model with lowest AIC
x.ar$ar # parameter estimate
sqrt(x.ar$asy.var) # asy. standard error
summary(x.ar)
x.ar$aic
```

4. The fourth command takes the square root of the asymptotic variance to arrive at the asymptotic standard error.
5. The fifth command lists different objects that come with the estimated AR object.
6. The last command displays the AIC.

Estimation: Example 1–AR(1)

The outputs of these commands follow:

```
> x.ar$order # order of the AR model with lowest AIC
[1] 1
> x.ar$ar # parameter estimate
[1] 0.6848115
> sqrt(x.ar$asy.var) # asy. standard error
          [,1]
[1,] 0.02304673
> summary(x.ar)
             Length Class  Mode
order           1   -none- numeric
ar              1   -none- numeric
var.pred        1   -none- numeric
x.mean          1   -none- numeric
aic            13   -none- numeric
n.used          1   -none- numeric
order.max       1   -none- numeric
partialacf      0   -none- NULL
resid        1000   ts     numeric
method          1   -none- character
series          1   -none- character
frequency       1   -none- numeric
call            3   -none- call
asy.var.coef    1   -none- numeric
> x.ar$aic
         0          1          2          3          4          5          6
630.074431   0.000000   1.945940   3.785605   4.626213   4.440357   5.231398
         7          8          9         10         11         12
  5.601880   7.464930   9.249846  11.194681  12.716578  14.715200
```

Estimation: Example 1–AR(1)

Note that the AICs associated with AR(0) to AR(12) models are all displayed. However, it is shown as the difference between the AIC of a model with the lowest AIC.

To get the AIC, we can calculate it "manually":

Akaike Information Criterion:

$$AIC = e^{\left(\frac{2k}{T}\right)} \frac{\sum_{t=1}^{T} e_t^2}{T}$$

- k is the number parameters
- n is the length of the sample used in the estimation
- MSE is mean squared error, which is a goodness-of-fit measure: the smaller the MSE, the better the fit.

Estimation: Example 1–AR(1)

Taking natural log transformation of the AIC formula above gives the following:

$$ln(AIC) = ln\left(\frac{\sum_{t=1}^{T} e_t^2}{T}\right) + \left(\frac{2k}{T}\right)$$
$$= ln(\text{MSE}) + \left(\frac{2k}{T}\right)$$

Some authors/statisticians simply call this "AIC" (instead of ln(AIC). Some other authors use other variants of AIC and called them AIC. For example, our textbook defines AIC as follows:

```
AIC = -2xlog-likelihood + 2k
```

Estimation: Example 1–AR(1)

Since the `ar()` function reports only the difference between each of the AICs and the lowest AIC, we will have to obtain AIC manually using what is provided by the `ar()` object.

The `ar()` object provides the value of the estimated maximum likelihood associated with the AR models with the lowest AIC, so we can use the following formula:

```
AIC = -2xlog-likelihood + 2k
```

```
> -2*(-1446.94) + 2*(3)
[1] 2899.88
```

Estimation: Example 1–AR(1)

Recall the several commands that we used to obtain the estimates from the model:

1. Order of the “best” AR model
2. Parameter estimates associated with it
3. Estimated (asymptotic) standard error
4. Confidence interval

The results are:

```
> x.ar$order # order of the AR model with lowest AIC
[1] 1
> x.ar$ar # parameter estimate
[1] 0.6848115
> sqrt(x.ar$asy.var) # asy. standard error
           [,1]
[1,] 0.02304673
> x.ar$ar + c(-2,2)*sqrt(x.ar$asy.var)
[1] 0.6387180 0.7309049
```

Estimation: Example 1–AR(1)

- Based on this particular realized sample path, the parameter estimates is `0.6748` and the **95% confidence interval** is `(0.6387, 0.7309)`, which includes the “true” parameter of the underlying data-generating process.

- In reality, however, the “true” parameter is unknown, no matter how big the sample is. In this example, we define the underlying data-generating process, so we know the “true” parameter. It just so happens that given this particular realized sample path, the estimated 95% confidence interval includes the true parameter.

Estimation: Example 1–AR(1)

- It is possible, however, that there are other sample paths whose associated confidence intervals do not include the true parameters.

- The interpretation of the 95% confidence interval is that given a large number of sample paths, 95% of those will include the true parameter.

- Try this as an exercise using the R scripts I've already provided. See the assignment for more detailed instructions.

Berkeley SCHOOL OF INFORMATION