Chapter 6, Part 2: Specification of Example Time Series

- We now use some of the tools we have learned to specify some actual time series (some simulated examples, some real examples).
- In R, we can obtain the ACF and the PACF and use these to decide between an AR model and an MA model, and to decide the correct order of the model.
- Sometimes neither a pure AR model nor a pure MA model seems to fit, so we could consider an ARMA model and use the EACF to determine the correct orders.
- We first consider some example simulated time series, which are generated to follow some known model, but we pretend we don't know the model and use our diagnostic tools to specify the model.

A Simulated $MA(1)$ Time Series

- See the example R code for the analysis of some simulated data (which actually follows an $MA(1)$ process).
- The ACF can be obtained in R using the `acf` function.
- By default, the `acf` function plots dotted lines at $\pm 2/\sqrt{n}$.
- This is two times the "naive" standard error – it would be correct under the white noise model.
- With the `ci.type='ma'` option, we obtain dotted lines at ± 2 times the more appropriate standard errors (which would be correct under the MA model).

The ACF of the Simulated $MA(1)$ Time Series

- First, we check whether the sample autocorrelations cut off (become not significantly different from zero) after a certain lag.
- If a sample autocorrelation remains inside the dotted lines, we can say it is not significantly different from zero.
- If the sample autocorrelations cut off after lag q, say, then this is evidence that an $MA(q)$ model may be correct.
- For our simulated data, can we conclude that an $MA(1)$ model is correct? Do the sample autocorrelations appear to "become zero" after lag 1?

Another Couple of Simulated *MA* Time Series

- In the R examples, we look at the ACF of a different simulated $MA(1)$ time series.
- Again, we check the ACF plot to see whether the sample autocorrelations cut off (become not significantly different from zero) after lag 1.
- The next example is a simulated $MA(2)$ time series.
- For this series, to determine whether it should be specified as an $MA(2)$ process, we check the ACF plot to see whether the sample autocorrelations cut off after lag 2.

Specification for AR Models

- In the next example, we consider a simulated $AR(1)$ process.
- Looking at the ACF, we see that the sample autocorrelations decay gradually toward zero, rather than cutting off abruptly after a certain lag.
- In fact, the sample ACF becomes negative from lags 10 through 16, which is surprising.
- The fact that the sample autocorrelations do not cut off after a certain lag tells us that we should NOT use an MA model for this time series.
- If we use an AR model for these data, we should examine the PACF to determine the order of the AR model.
- In this simulated example, the PACF cuts off after lag 1, so an $AR(1)$ model makes sense.

An Example of a Simulated $AR(2)$ Data Set

- The next R example is from a simulated time series that actually follows an $AR(2)$ process.
- The ACF for this series does display the "wave" shape that we might expect from an $AR(2)$ process.
- The PACF shows a cutoff after lag 2, so an $AR(2)$ model makes sense.

Simulating an $ARMA(1,1)$ Data Set

- Exhibit 6.14 on page 123 shows a simulated time series following the $ARMA(1,1)$ process.
- Based on the ACF and especially the PACF, an $AR(1)$ model would seem reasonable for this time series.
- But the EACF (obtained with the `eacf` function in R) indicates that an $ARMA(1,1)$ or $ARMA(2,1)$ model would fit well.
- We see that these diagnostic tools do not always give clearcut answers as to which model is best.
- This is true even with simulated data, so with real data (which might only approximately follow a common model), the performance of these diagnostic tools might be even more shaky.

Assessing Nonstationarity through ACFs

- We have seen that many real time series exhibit *nonstationary* behavior.
- For these, ARIMA would be a better model than ARMA-type models.
- Sometimes the nonstationarity can be seen from a regular plot of the time series: for example, if we can see that the mean or the variance changes over time.
- Sometimes the ACF plot can reveal nonstationarity as well.
- With a nonstationary series, the ACF typically does not die off quickly as the lag increases.

Example of Assessing Nonstationarity through ACFs

- With the oil price data, the ACF clearly does not die off, and all the displayed sample autocorrelations are significantly nonzero.
- Taking a log transformation of the oil prices and then taking first differences, we see the ACF supports a $MA(1)$ model.
- That would indicate that the original data (or a logged version of it) could follow an $IMA(1, 1)$ process.
- If taking the first differences of a nonstationary series does not produce a series that appears stationary, then we could try taking the second differences and examining the ACF and PACF.

Overdifferencing

- In a time series is already stationary, then its differences will also be stationary.
- But we should *not* take differences on a series that is already stationary.
- For example, if the original data follow a random walk process, then taking first differences would produce a stationary *white noise* model.
- If we take second differences (this would be *overdifferencing*), it theoretically produces an $MA(1)$ process with $\theta = 1$ (though with an actual data set we would be forced to estimate θ).
- But to claim the original series is $IMA(2, 1)$ is incorrect: The random walk is actually an $IMA(1, 1)$ process with $\theta = 0$.

More on Overdifferencing

- Overdifferencing will create a noninvertible model, which leads to problems with interpretability and parameter estimation.
- See the R example with a simulated random walk series to illustrate the dangers of overdifferencing.
- To prevent overdifferencing, it is recommended to look carefully at each difference in succession and not to choose a model that is more complicated than necessary.

The Dickey-Fuller Unit Root Test

- The *Dickey-Fuller Unit Root Test* is a formal hypothesis test for whether the time series is "difference nonstationary."
- The null hypothesis is that the series is nonstationary, but can be made stationary by differencing.
- The alternative hypothesis is that the series is stationary.
- The assumption of the test is that the time series follows an $AR(k)$ process, but in practice k is unknown, and must be estimated.
- The test statistic of the augmented Dickey-Fuller (ADF) test is a t-statistic from a least squares regression of the first differences of $\{Y_t\}$ on the lag-1 of the series and the past k lags of the first-differenced series.

More on the Dickey-Fuller Unit Root Test

- In other cases, we may wish to test for "trend nonstationarity," which implies that the series has a deterministic trend, but is stationary once this trend is removed.
- This can be tested by performing the ADF test on the detrended data, or equivalently by including the covariates defining the trend in the previously mentioned regression.
- The R function `adfTest` in the `fUnitRoots` package can perform these ADF tests for difference nonstationarity and for trend nonstationarity.
- The R function `adf.test` in the `tseries` package can perform the ADF test for trend nonstationarity.
- See the R examples on the course web page.

Other Methods of Specification: The AIC

- A general method of model selection is to choose the model with the smallest *Akaike Information Criterion* (AIC):

$$AIC = -2 \log L + 2k,$$

  where L here is the maximized likelihood function and $k = p + q + 1$ for a model with an intercept term and $k = p + q$ for a model without an intercept.
- Models with a large L have a good fit to the data, while models with a small k are less complex, so the $2k$ piece serves as a "penalty" that discourages choosing overly complex models, even if those models produce a good fit to the observed data.

Other Methods of Specification: More on the AIC

- The AIC can be viewed as a (biased) estimator of the "Kullback-Leibler divergence" of the estimated model from the true model.
- A bias-corrected version of the AIC,

$$AIC_c = AIC + \frac{2(k+1)(k+2)}{n-k-2}$$

is occasionally used as a criterion, since it is an unbiased estimator of the Kullback-Leibler divergence.

Another Method of Specification

- Another common method of model selection is to choose the model with the smallest *Bayesian Information Criterion* (BIC):

$$BIC = -2 \log L + k \log(n),$$

  which (similar to AIC) is a penalized measure of goodness of fit.
- If the true model is $ARMA(p, q)$, the BIC has the nice property of consistency: As the sample size gets larger, the estimates of p and q produced using the BIC approach the true orders p and q.
- If the true model is not $ARMA(p, q)$, then using AIC can produce the estimated ARMA process that is closest possible to the true process, among a large class of ARMA models.

Issues with Estimation

- Using either AIC or BIC requires maximum likelihood estimation, which can lead to numerical problems with ARMA models due to the complicated likelihood function.
- The Hanna-Rissanen approach to estimating ARMA models consists of two steps:
  1. Fitting a high-order AR process and determining the correct order by minimizing AIC;
  2. Then estimating k and j of an $ARMA(k, j)$ model by regressing the time series on its own lags 1 to k and on lags 1 to j of the residuals from the high-order AR model.
- The best approach for estimating the orders of an $ARMA(p, q)$ model is to consider several "best subset" models that may include a few lag terms in them.
- The R function *armasubsets* produces a summary of some "best" ARMA models; see the example on some simulated $ARMA(12, 12)$ data.

Specification of Some Actual Time Series

- Consider the Los Angeles rainfall data (recall that an exploratory analysis of this time series did not reveal any notable year-to-year dependence).
- We see taking logarithms of the data makes the responses more normally distributed.
- The sample ACF of the log-transformed data shows no dependence evident at *any* lag.
- It would be sensible to model these values as independent (actually, iid) random variables over time.

Specification of the Color Property Time Series

- Exploratory plots of the color property series did indicate some association between color values in successive batches.
- The sample ACF shows significant autocorrelation at lag-1; should we consider an $MA(1)$ model?
- Note the "damped sine wave" appearance of the sample ACF, however, which encourages us to examine the sample PACF.
- The sample PACF shows a significant partial autocorrelation at lag 1, and near zero sample partial autocorrelations at other lags.
- Based on this, an $AR(1)$ model may be most appropriate, but we will investigate further using model diagnostics.

Specification of the Canadian Hare Time Series

- The Canadian hare abundance series also showed signs of dependence over time.
- Could a transformation of the abundance values improve the modeling of these data?
- A Box-Cox analysis suggests a square-root transformation.
- The sample ACF again shows the damped sine wave pattern.
- The sample PACF supports an $AR(2)$ model (or maybe $AR(3)$?).

Specification of the Oil Price Time Series

- For the oil price time series, there was graphical evidence that the differenced logarithms of the oil prices were stationary.
- The augmented Dickey-Fuller test concludes the logged oil price series itself is nonstationary (large P-value).
- So differencing the logged series makes sense.
- An EACF table of the differences of the logged prices can indicate appropriate orders p and q for an $ARMA(p, q)$ model.
- The table suggests the choices $p = 0$ and $q = 1$ may work well.

Specification of the Oil Price Time Series, Continued

- The `armasubsets` function in the `TSA` package, applied to the differences of logs of oil prices, suggests including Y_{t-1} and Y_{t-4}, and no lags in the error terms.
- The next best model includes Y_{t-1} (and again no lags in the error terms), which corresponds to an $ARIMA(1, 1, 0)$ model for the logged oil price series itself.
- The sample ACF possibly suggests an $MA(1)$ model (but is there a damped sine wave pattern?).
- The sample PACF suggests an $AR(2)$ model (although note the large spikes at later lags).
- We could consider all of these models (possibly using an overall criterion like AIC?) when we undertake parameter estimation and model diagnostics.
- Outliers in the oil price time series should also be dealt with (see original time series plot).

Specification of Other Time Series

- The ACF and PACF of the recruitment data shows a recognizable pattern.
- The ACF and PACF of the SOI data do not appear to correspond to any stationary ARMA model.
- Does taking first differences yield a recognizable model?
- If not, we could try second differences.
- If we take a log transformation of the Johnson and Johnson earnings data, we get an ACF and PACF with a recognizable pattern.
- Sometimes the patterns shown by the ACF, PACF, and EACF indicate something more complicated than a simple AR or MA model; see the airmiles example.

Using the `auto.arima` Function for Automated Model Selection

- The `auto.arima` function in the `forecast` package automatically searches among a large class of ARIMA models and picks the one with the lowest AIC (or BIC or AIC_c, if the user wants).
- By default, it uses a stepwise model selection approach to make the search faster.
- The function considers values of p and q up to 5 and values of d up to 2 (these can be adjusted with the arguments `max.p`, `max.q`, and `max.d`).

Caution About Comparing ARMA vs. ARIMA models using *AIC*

- One should not compare an ARMA model (with $d = 0$) to an ARIMA model (with $d > 0$) using *AIC* or any other information criterion.
- Since these two models have different response variables (one uses Y_t and the other uses $\nabla^d Y_t$), the *AIC* values for these models are not comparable.
- The correct amount of differencing (if any) should be chosen first, and then *AIC* can be used to guide the choices of p and/or q.
- The `auto.arima` function first decides on the best value of d (using a unit root test) and then chooses the best values of p and q based on the specified information criterion (such as *AIC*).

Caveats About the `auto.arima` Function

- The `auto.arima` function is quick and useful, especially if many ARIMA fits must be done in an automated way.
- But when analyzing a single series, using the `auto.arima` function should not replace a complete investigation of the behavior of the series (such as via the ACF, PACF and/or EACF).
- See the R examples for using `auto.arima` on some real data examples.
- By default, `auto.arima` also considers some more complicated models that we will study in Chapter 9 and 10.