17

STAT 626: Outline of Lecture ~~22~~

The ARIMA (p, d, q) Model Building Process (§5.2)

1. **Plot the Data, Transform to Stationarity if Necessary,**
   Select the Differencing Order d.

2. **Model Formulation: Use the ACF and PACF to Select p, q:**
   ARIMA(p, d, q).

3. **Model Estimation: Find the MLE of the $p + q + 1$ Parameters**

4. **Model Diagnostic: Check the Residuals for Independence**

5. **If Not Happy, Go to Step 2 and Repeat the PROCESS**

6. **Choose from the Competing Models Using AIC/BIC**

7. Review of HWs on the NEGATIVE role of NONSTATIONARITY.

Example 5.6: Analysis of GNP Data

Example 5.8 Diagnostics for the Glacial Varve Series

ALL Models Are Wrong, But SOME Are Useful.

Who said the above?

STAT 626: Review of Past Lectures

1. Forecating: Begins when a good model is identified for the time series,

2. Given the time series data $x_1, \ldots, x_n$: **What are the principles for model-based forecasting ?**

$$x_t = f(\beta, \text{Past of the Series}) + w_t.$$

   Example:

$$x_t = \phi x_{t-1} + w_t.$$

   Principle: Replace the unknowns by the best ESTIMATES.
   Example:

$$x_t = w_t + \theta w_{t-1}.$$

3. **Forecasting ARMA Models**

   Recall that causal ARMA models can be written as One-Sided MA(∞) of a white noise:

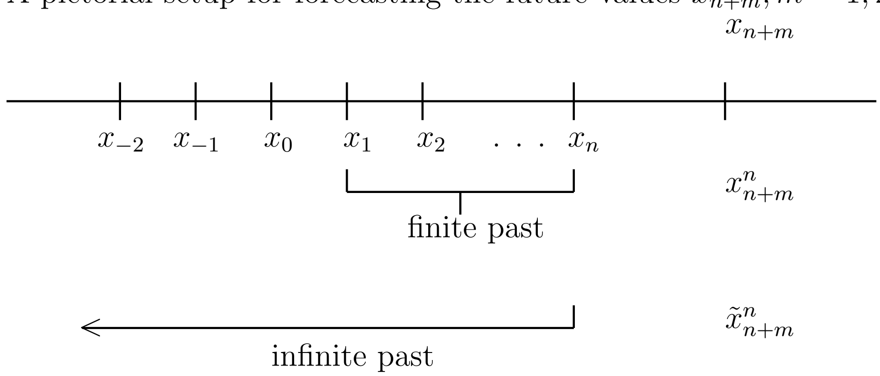
$$x_t = \sum_{j=0}^{\infty} \psi_j w_{t-j},$$

   and **invertible** ARMA models can be written as One-Sided AR(∞);

$$x_t = \sum_{j=0}^{\infty} \pi_j x_{t-j} + w_t.$$

Forecasting

4. A pictorial setup for forecasting the future values $x_{n+m}, m = 1, 2, \ldots$:



5. What are their forecasts, forecast error, and forecast error variances?

Forecast error: $x_{n+m} - x^n_{n+m}$

Error variance: $P^n_{n+m} = \mathrm{Var}(x_{n+m} - x^n_{n+m})$

6. Their 95% forecast intervals?

$$x^n_{n+m} \pm 1.96\sqrt{P^n_{n+m}}.$$

Chapter 5

ARIMA Models

5.1 Integrated Models

Adding nonstationary to ARMA models leads to the *autoregressive integrated moving average* (ARIMA) model popularized by Box and Jenkins (1970). Seasonal data, such as the data discussed in Example 1.1 and Example 1.4 lead to seasonal autoregressive integrated moving average (SARIMA) models.

In previous chapters, we saw that if x_t is a random walk, $x_t = x_{t-1} + w_t$, then by differencing x_t, we find that $\nabla x_t = w_t$ is stationary. In many situations, time series can be thought of as being composed of two components, a nonstationary trend component and a zero-mean stationary component. For example, in Section 3.1 we considered the model

$$x_t = \mu_t + y_t, \tag{5.1}$$

where $\mu_t = \beta_0 + \beta_1 t$ and y_t is stationary. Differencing such a process will lead to a stationary process:

$$\nabla x_t = x_t - x_{t-1} = \beta_1 + y_t - y_{t-1} = \beta_1 + \nabla y_t.$$

Another model that leads to first differencing is the case in which μ_t in (5.1) is stochastic and slowly varying according to a random walk. That is,

$$\mu_t = \mu_{t-1} + v_t$$

where v_t is stationary and uncorrelated with y_t. In this case,

$$\nabla x_t = v_t + \nabla y_t,$$

is stationary.

On a rare occasion, the differenced data ∇x_t may still have linear trend or random walk behavior. In this case, it may be appropriate to difference the data again, $\nabla(\nabla x_t) = \nabla^2 x_t$. For example, if μ_t in (5.1) is quadratic, $\mu_t = \beta_0 + \beta_1 t + \beta_2 t^2$, then the twice differenced series $\nabla^2 x_t$ is stationary.

The *integrated* ARMA, or ARIMA, model is a broadening of the class of ARMA models to include differencing. The basic idea is that if differencing the data at some order d produces an ARMA process, then the original process is said to be ARIMA. Recall that the difference operator defined in

Definition 5.1. *A process x_t is said to be* **ARIMA**(p,d,q) *if*

$$\nabla^d x_t = (1-B)^d x_t$$

is ARMA(p,q). In general, we will write the model as

$$\phi(B)(1-B)^d x_t = \alpha + \theta(B) w_t, \tag{5.2}$$

where $\alpha = \delta(1-\phi_1 - \cdots - \phi_p)$ and $\delta = E(\nabla^d x_t)$.

Estimation for ARIMA models is the same as for ARMA models except that the data are differenced first. For example, if $d=1$, we fit an ARMA model to $\nabla x_t = x_t - x_{t-1}$ instead of x_t.

Example 5.2. Fitting the Glacial Varve Series (cont.)
In Example 4.28, we fit an MA(1) to the differenced logged varve series as using the commands as follows:

```
sarima(diff(log(varve)), p=0, d=0, q=1, no.constant=TRUE)
```

Equivalently, we can fit an ARIMA$(0,1,1)$ to the logged series:

```
sarima(log(varve), p=0, d=1, q=1, no.constant=TRUE)
 Coefficients:
          ma1
      -0.7705
 s.e.   0.0341
 sigma^2 estimated as 0.2353
```

The results are identical to Example 4.28. The only difference will be when we forecast. In Example 4.28 we would get forecasts of $\nabla \log x_t$ and in this example we would get forecasts for $\log x_t$, where x_t represents the varve series. ◊

Forecasting ARIMA is also similar to the ARMA case, but needs some additional consideration. Since $y_t = \nabla^d x_t$ is ARMA, we can use Section 4.4 methods to obtain forecasts of y_t, which in turn lead to forecasts for x_t. For example, if $d=1$, given forecasts y^n_{n+m} for $m=1,2,\ldots$, we have $y^n_{n+m} = x^n_{n+m} - x^n_{n+m-1}$, so that

$$x^n_{n+m} = y^n_{n+m} + x^n_{n+m-1}$$

with initial condition $x^n_{n+1} = y^n_{n+1} + x_n$ (noting $x^n_n = x_n$).

It is a little more difficult to obtain the prediction errors P^n_{n+m}, but for large n, the approximation (4.39) works well. That is, the mean-squared prediction error (MSPE) can be approximated by

$$P^n_{n+m} = \sigma^2_w \sum_{j=0}^{m-1} \psi_j^2, \tag{5.3}$$

where ψ_j is the coefficient of z^j in $\psi(z) = \theta(z)/\phi(z)(1-z)^d$; Section D.2 has more details on how the ψ-weights are determined.

To better understand forecasting integrated models, we examine the properties of some simple cases.

Example 5.3. Random Walk with Drift
To fix ideas, we begin by considering the random walk with drift model first presented in Example 1.10, that is,

$$x_t = \delta + x_{t-1} + w_t,$$

for $t = 1,2,\ldots$, and $x_0 = 0$. Given data $x_1,\ldots,x_n$, the one-step-ahead forecast is given by

$$x^n_{n+1} = \delta + x^n_n + w^n_{n+1} = \delta + x_n\,.$$

The two-step-ahead forecast is given by $x^n_{n+2} = \delta + x^n_{n+1} = 2\delta + x_n$, and consequently, the m-step-ahead forecast, for $m = 1,2,\ldots$, is

$$x^n_{n+m} = m\,\delta + x_n, \tag{5.4}$$

To obtain the forecast errors, it is convenient to recall equation (1.4) wherein $x_n = n\,\delta + \sum_{j=1}^n w_j$. In this case we may write

$$x_{n+m} = (n+m)\,\delta + \sum_{j=1}^{n+m} w_j = m\,\delta + x_n + \sum_{j=n+1}^{n+m} w_j\,. \tag{5.5}$$

Using the difference of (5.5) and (5.4), it follows that the m-step-ahead prediction error is given by

$$P^n_{n+m} = E(x_{n+m} - x^n_{n+m})^2 = E\Big(\sum_{j=n+1}^{n+m} w_j\Big)^2 = m\,\sigma^2_w. \tag{5.6}$$

Unlike the stationary case, as the forecast horizon grows, the prediction errors, (5.6), increase without bound and the forecasts follow a straight line with slope δ emanating from x_n.

We note that (5.3) is exact in this case because the ψ-weights for this model are all equal to one. Thus, the MSPE is $P^n_{n+m} = \sigma^2_w \sum_{j=0}^{m-1}\psi_j^2 = m\sigma^2_w$.

```
ARMAtoMA(ar=1, ma=0, 20)  # ψ-weights
 [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

◊

Example 5.4. Forecasting an ARIMA(1,1,0)
To get a better idea of what forecasts for ARIMA models will look like, we generated 150 observations from an ARIMA(1, 1, 0) model,

$$\nabla x_t = .9 \nabla x_{t-1} + w_t\,.$$

Alternately, the model is $x_t - x_{t-1} = .9(x_{t-1} - x_{t-2}) + w_t$, or

$$x_t = 1.9 x_{t-1} - .9 x_{t-2} + w_t\,.$$

Although this form looks like an AR(2), the model is not causal in x_t and therefore not an AR(2). As a check, notice that the ψ-weights do not converge to zero (and in fact converge to 10).

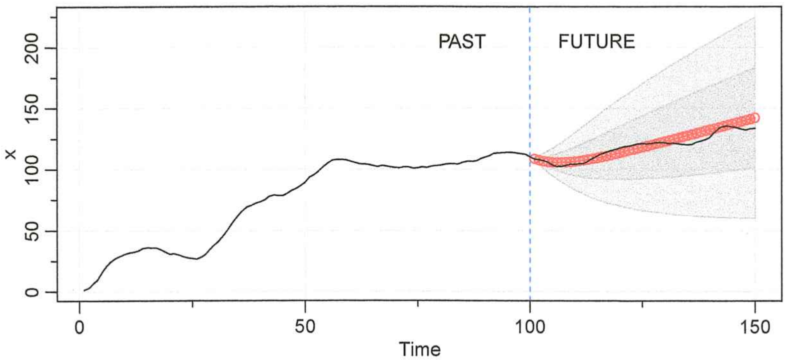


Figure 5.1 *Output for* Example 5.4*: Simulated ARIMA*(1, 1, 0) *series (solid line) with out of sample forecasts (points) and error bounds (gray area) based on the first 100 observations.*

```
round( ARMAtoMA(ar=c(1.9,-.9), ma=0, 60), 1 )
 [1]  1.9  2.7  3.4  4.1  4.7  5.2  5.7  6.1  6.5  6.9  7.2  7.5
[13]  7.7  7.9  8.1  8.3  8.5  8.6  8.8  8.9  9.0  9.1  9.2  9.3
[25]  9.4  9.4  9.5  9.5  9.6  9.6  9.7  9.7  9.7  9.7  9.8  9.8
[37]  9.8  9.8  9.9  9.9  9.9  9.9  9.9  9.9  9.9  9.9  9.9  9.9
[49]  9.9 10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0
```

We used the first 100 (of 150) generated observations to estimate a model and then predicted out-of-sample, 50 time units ahead. The results are displayed in Figure 5.1 where the solid line represents all the data, the points represent the forecasts, and the gray areas represent ± 1 and ± 2 root MSPEs. Note that, unlike the forecasts of an ARMA model from the previous chapter, the error bounds continue to increase.

The R code to generate Figure 5.1 is below. Note that `sarima.for` fits an ARIMA model and then does the forecasting out to a chosen horizon. In this case, `x` is the entire time series of 150 points, whereas `y` is only the first 100 values of `x`.

```
set.seed(1998)
x <- ts(arima.sim(list(order = c(1,1,0), ar=.9), n=150)[-1])
y <- window(x, start=1, end=100)
sarima.for(y, n.ahead = 50, p = 1, d = 1, q = 0, plot.all=TRUE)
text(85, 205, "PAST"); text(115, 205, "FUTURE")
abline(v=100, lty=2, col=4)
lines(x)
```

◊

Example 5.5. IMA(1,1) and EWMA

The ARIMA(0,1,1), or IMA(1,1) model is of interest because many economic time series can be successfully modeled this way. The model leads to a frequently used method called exponentially weighted moving average (EWMA). We will write the model as

$$x_t = x_{t-1} + w_t - \lambda w_{t-1}, \tag{5.7}$$

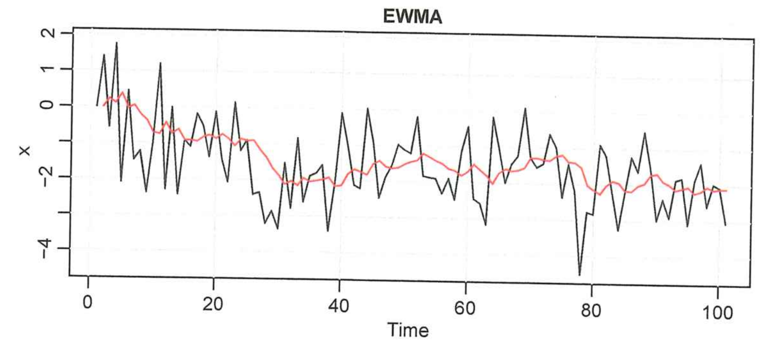


Figure 5.2 *Output for* Example 5.5*: Simulated data with an EWMA superimposed.*

with $|\lambda| < 1$, because this model formulation is easier to work with here, and it leads to the standard representation for EWMA.

In this case, the one-step-ahead predictor is

$$x_{n+1}^n = (1-\lambda)x_n + \lambda x_n^{n-1}. \tag{5.8}$$

That is, the predictor is a linear combination of the present value of the process, x_n, and the prediction of the present, x_n^{n-1}. Details are given in Problem 5.17. This method of forecasting is popular because it is easy to use; we need only retain the previous forecast value and the current observation to forecast the next time period. EWMA is widely used, for example in control charts (Shewhart, 1931), and economic forecasting (Winters, 1960) whether or not the underlying dynamics are IMA(1,1).

The MSPE is given by

$$P_{n+m}^n \approx \sigma_w^2[1 + (m-1)(1-\lambda)^2]. \tag{5.9}$$

In EWMA, the parameter $1-\lambda$ is often called the smoothing parameter, is denoted by α, and is restricted to be between zero and one. Larger values of λ (or smaller values of α) lead to smoother forecasts.

In the following, we show how to generate 100 observations from an IMA(1,1) model with $\alpha = 1-\lambda = .2$ and then calculate and display the fitted EWMA superimposed on the data. This can accomplished using the Holt-Winters command in R (see the help file `?HoltWinters` for details). This and related techniques are generally called *exponential smoothing*; the ideas were made popular in the late 1950s and are still used today. To reproduce Figure 5.2, use the following.

```
set.seed(666)
x = arima.sim(list(order = c(0,1,1), ma = -0.8), n = 100)
(x.ima = HoltWinters(x, beta=FALSE, gamma=FALSE))  # α below is 1−λ
  Smoothing parameter:  alpha:   0.1663072
plot(x.ima, main="EWMA")
```

◊

5.2 Building ARIMA Models

There are a few basic steps to fitting ARIMA models to time series data. These steps involve

- plotting the data,
- possibly transforming the data,
- identifying the dependence orders of the model,
- parameter estimation,
- diagnostics, and
- model choice.

First, as with any data analysis, construct a time plot of the data and inspect the graph for any anomalies. It may be of interest to transform the data and as we have seen in numerous examples, if the data behave as $x_t = (1 + r_t)x_{t-1}$, where r_t is a stable process of small percent changes, then $\nabla \log(x_t) \approx r_t$ will be stable. This general idea was used in Example 4.27, and we will use it again in Example 5.6.

After suitably transforming the data, the next step is to identify preliminary values of the autoregressive order, p, the order of differencing, d, and the moving average order, q. A time plot of the data will typically suggest whether any differencing is needed. If differencing is called for, then difference the data once, $d = 1$, and inspect the time plot of ∇x_t. If additional differencing is necessary, then try differencing again and inspect a time plot of $\nabla^2 x_t$; *it is rare for d to be bigger than 1*. Be careful not to overdifference because this may introduce dependence where none exists. For example, $x_t = w_t$ is serially uncorrelated, but $\nabla x_t = w_t - w_{t-1}$ is a non-invertible MA(1). In addition to time plots, the sample ACF can help in indicating whether differencing is needed. A slow (linear) decay in the ACF is an indication that differencing may be needed.

When preliminary values of d have been chosen (including no differencing, $d = 0$), the next step is to look at the sample ACF and PACF of $\nabla^d x_t$. Using Table 4.1 as a guide, preliminary values of p and q are chosen. Note that it cannot be the case that both the ACF and PACF cut off. Because we are dealing with estimates, it will not always be clear whether the sample ACF or PACF is tailing off or cutting off. Also, two models that are seemingly different can actually be very similar. It is a good idea to *start small* and up the orders slowly. Also, watch out for parameter redundancy and do not increase p and q at the same time. At this point, a few preliminary values of p, d, and q should be at hand, and we can start estimating the parameters and performing diagnostics and model choice.

Example 5.6. Analysis of GNP Data

In this example, we consider the analysis of quarterly U.S. GNP from 1947(1) to 2002(3), $n = 223$ observations. The data are real U.S. gross national product in billions of chained 1996 dollars and have been seasonally adjusted. Figure 5.3 shows a plot of the data, say, y_t. Because strong trend tends to obscure other effects, it is difficult to see any other variability in data except for periodic large dips in

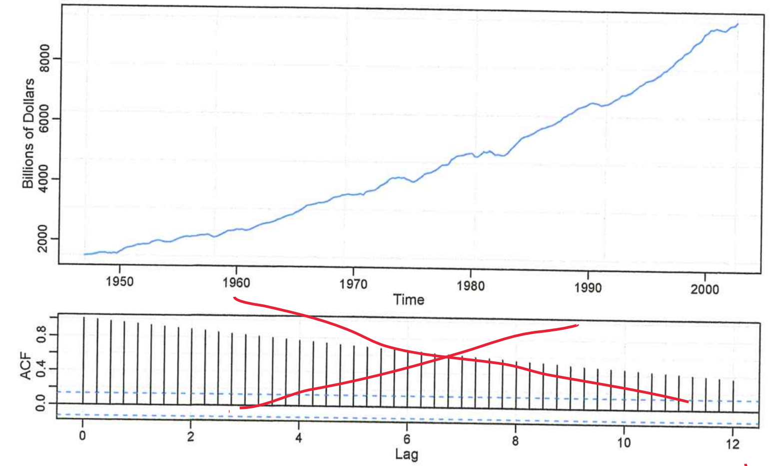


Figure 5.3 Top: *Quarterly U.S. GNP from 1947(1) to 2002(3).* Bottom: *Sample ACF of the GNP data. Lag is in terms of years.*

the economy. Typically, GNP and similar economic indicators are given in terms of growth rate (percent change) rather than in actual values. The growth rate, say $x_t = \nabla \log(y_t)$, is plotted in Figure 5.4 and it appears to be a stable process.

```
##-- Figure 5.3 --##
layout(1:2, heights=2:1)
tsplot(gnp, col=4)
acf1(gnp, main="")
##-- Figure 5.4 --##
tsplot(diff(log(gnp)), ylab="GNP Growth Rate", col=4)
abline(mean(diff(log(gnp))), col=6)
##-- Figure 5.5 --##
acf2(diff(log(gnp)), main="")
```

The sample ACF and PACF of the quarterly growth rate are plotted in Figure 5.5. Inspecting the sample ACF and PACF, we might feel that the ACF is cutting off at lag 2 and the PACF is tailing off. This would suggest the GNP growth rate follows an MA(2) process, or log GNP follows an ARIMA$(0,1,2)$ model.

The MA(2) fit to the growth rate, x_t, is

$$\hat{x}_t = .008_{(.001)} + .303_{(.065)}\hat{w}_{t-1} + .204_{(.064)}\hat{w}_{t-2} + \hat{w}_t, \tag{5.10}$$

where $\hat{\sigma}_w = .0094$ is based on 219 degrees of freedom.

```
sarima(diff(log(gnp)), 0,0,2)    # MA(2) on growth rate
        Estimate     SE t.value p.value
  ma1     0.3028 0.0654  4.6272  0.0000
  ma2     0.2035 0.0644  3.1594  0.0018
  xmean   0.0083 0.0010  8.7178  0.0000
  sigma^2 estimated as 8.919e-05
```

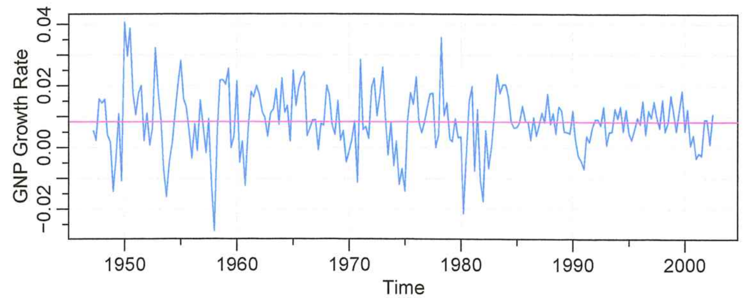


Figure 5.4 *U.S. GNP quarterly growth rate. The horizontal line displays the average growth of the process, which is close to 1%.*

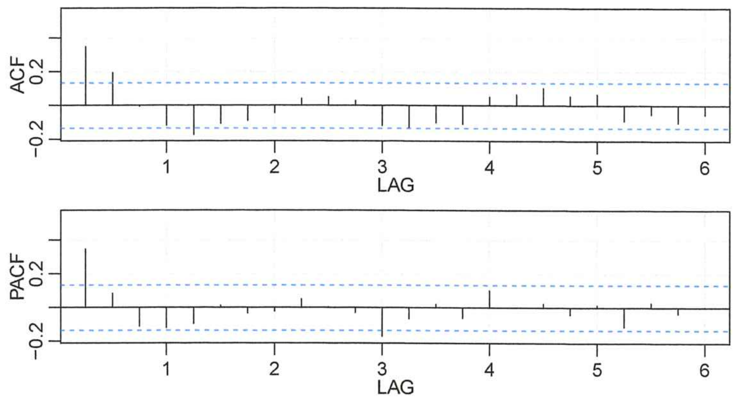


Figure 5.5 *Sample ACF and PACF of the GNP quarterly growth rate. Lag is in years.*

We note that `sarima(log(gnp), p=0, d=1, q=2)` will produce the same results.

All of the regression coefficients are significant, including the constant. *We make a special note of this because, as a default, some computer packages—including the* R *stats package—do not fit a constant in a differenced model, assuming without reason that there is no drift.* In this example, not including a constant leads to the wrong conclusions about the nature of the U.S. economy. Not including a constant assumes the average quarterly growth rate is zero, whereas the U.S. GNP average quarterly growth rate is about 1% (which can be seen easily in Figure 5.4).

Rather than focus on one model, we will also suggest that it appears that the ACF is tailing off and the PACF is cutting off at lag 1. This suggests an AR(1) model for the growth rate, or ARIMA$(1,1,0)$ for log GNP. The estimated AR(1) model is

$$\hat{x}_t = .008_{(.001)}\,(1 - .347) + .347_{(.063)}\,x_{t-1} + \hat{w}_t, \tag{5.11}$$

where $\hat{\sigma}_w = .0095$ on 220 degrees of freedom; note that the constant in (5.11) is $.008\,(1 - .347) = .005$.


```
sarima(diff(log(gnp)), 1,0,0)     # AR(1) on growth rate
          Estimate     SE t.value p.value
   ar1      0.3467 0.0627  5.5255       0
   xmean    0.0083 0.0010  8.5398       0
   sigma^2 estimated as 9.03e-05
```

As before, `sarima(log(gnp), p=1, d=1, q=0)` will produce the same results.

We will discuss diagnostics next, but assuming both of these models fit well, how are we to reconcile the apparent differences of the estimated models (5.10) and (5.11)? In fact, the fitted models are nearly the same. To show this, consider an AR(1) model of the form in (5.11) without a constant term; that is,

$$x_t = .35x_{t-1} + w_t,$$

and write it in its causal form, $x_t = \sum_{j=0}^{\infty} \psi_j w_{t-j}$, where we recall $\psi_j = .35^j$. Thus, $\psi_0 = 1, \psi_1 = .350, \psi_2 = .123, \psi_3 = .043, \psi_4 = .015, \psi_5 = .005, \psi_6 = .002, \psi_7 = .001, \psi_8 = 0, \psi_9 = 0, \psi_{10} = 0$, and so forth. The AR(1) model is approximately an MA(2) model,

$$x_t \approx .35w_{t-1} + .12w_{t-2} + w_t,$$

which is similar to the fitted MA(2) model in (5.10).

```
round( ARMAtoMA(ar=.35, ma=0, 10), 3)   # print psi-weights
 [1] 0.350 0.122 0.043 0.015 0.005 0.002 0.001 0.000 0.000 0.000
```

◇

The next step in model fitting is residual diagnostics. The first step involves a time plot of the *innovations* (or residuals), $x_t - \hat{x}_t^{t-1}$, or of the *standardized innovations*

$$e_t = \left(x_t - \hat{x}_t^{t-1}\right) \Big/ \sqrt{\hat{P}_t^{t-1}}, \tag{5.12}$$

where $\hat{x}_t^{t-1}$ is the one-step-ahead prediction of x_t based on the fitted model and $\hat{P}_t^{t-1}$ is the estimated one-step-ahead error variance. If the model fits well, the standardized residuals should behave as an independent normal sequence with mean zero and variance one. The time plot should be inspected for any obvious departures from this assumption. Investigation of marginal normality can be accomplished visually by inspecting a normal Q-Q plot.

We should also inspect the sample autocorrelations of the residuals, say $\hat{\rho}_e(h)$, for any patterns or large values. In addition to plotting $\hat{\rho}_e(h)$, we can perform a general test of whiteness that takes into consideration the magnitudes of $\hat{\rho}_e(h)$ as a group. The *Ljung–Box–Pierce Q-statistic* given by

$$Q = n(n+2)\sum_{h=1}^{H} \frac{\hat{\rho}_e^2(h)}{n-h} \tag{5.13}$$

can be used to perform such a test. The value H in (5.13) is chosen somewhat arbitrarily, but not too large. For large sample sizes, under the null hypothesis of model adequacy $Q \stackrel{\cdot}{\sim} \chi^2_{H-p-q}$. Thus, we would reject the null hypothesis at level α if the value of Q exceeds the $(1-\alpha)$-quantile of the χ^2 distribution

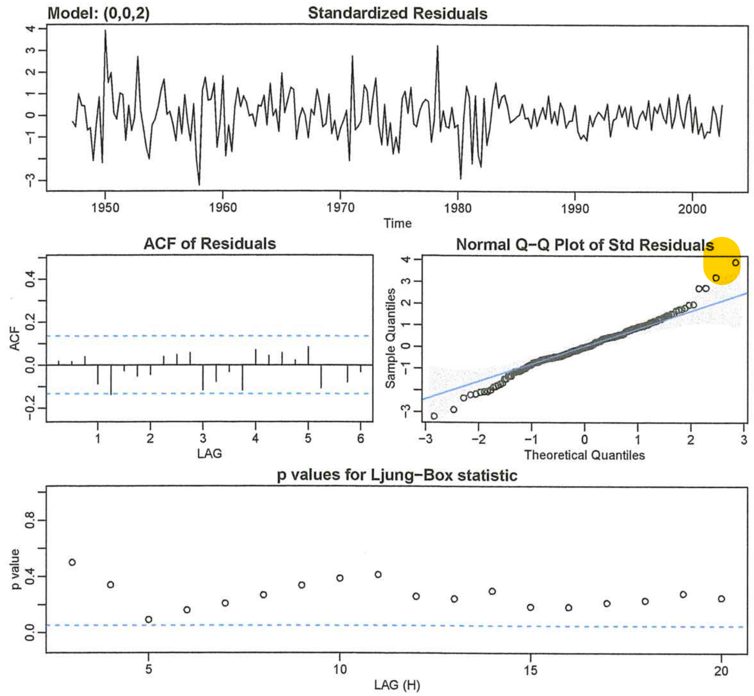


Figure 5.6 *Diagnostics of the residuals from MA(2) fit on GNP growth rate.*

Example 5.7. Diagnostics for GNP Growth Rate Example

We will focus on the MA(2) fit from Example 5.6; the analysis of the AR(1) residuals is similar. Figure 5.6 displays a plot of the standardized residuals, the ACF of the residuals, a Q-Q plot of the standardized residuals, and the p-values associated with the Q-statistic, (5.13). The residual analysis figure is generated as part of the call:

```
sarima(diff(log(gnp)), 0, 0, 2)   # MA(2) fit with diagnostics
```

You can turn off the diagnostics by adding `details=FALSE` in the `sarima` call.

Inspection of the time plot of the standardized residuals in Figure 5.6 shows no obvious patterns. Notice that there may be outliers because a few standardized residuals exceed 3 standard deviations in magnitude. However, there are no values that are exceedingly large in magnitude.

The ACF of the residuals shows no apparent departure from the model assumptions. The normal Q-Q plot of the residuals suggests that the assumption of normality is not unreasonable, however, there may be one large positive outlier.

Next, consider the Q-statistic. The graphic shows the p-values for the tests based on the lags $H = 3$ through $H = 20$ (with corresponding degrees of freedom $H - 2$). The dashed horizontal line on the bottom indicates the .05 level. The way to view this graphic is not as doing 17 highly dependent tests, but as another way to view the ACF of the residuals. In particular, the Q-statistic looks at the accumulation

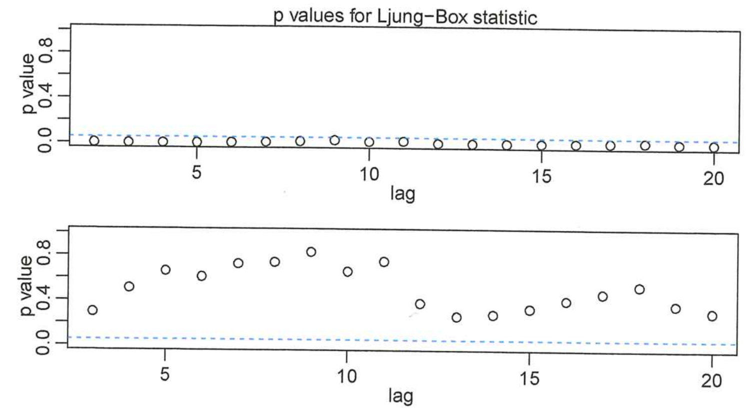


Figure 5.7 *Q-statistic p-values for the ARIMA$(0,1,1)$ fit (top) and the ARIMA$(1,1,1)$ fit (bottom) to the logged varve data.*

of autocorrelation rather than individual autocorrelations seen in the ACF. In this example all the p-values exceed .05, so we can feel comfortable not rejecting the null hypothesis that the residuals are white.

As a final check, we might consider overfitting a model to see if the results change significantly. For example, we might try the following,

```
sarima(diff(log(gnp)), 0, 0, 3)   # try an MA(2+1) fit (not shown)
sarima(diff(log(gnp)), 2, 0, 0)   # try an AR(1+1) fit (not shown)
```

and conclude that the extra parameter does not significantly change the results. ◊

Example 5.8. Diagnostics for the Glacial Varve Series

In Example 5.2, we fit an ARIMA$(0,1,1)$ model to the logarithms of the glacial varve data and there appears to be a small amount of autocorrelation left in the residuals and the Q-tests are all significant; see Figure 5.7.

To adjust for the small amount of autocorrelation left by the model, we added an AR parameter to the mix and fit an ARIMA$(1,1,1)$ to the logged varve data.

```
sarima(log(varve), 0, 1, 1, no.constant=TRUE)   # ARIMA(0,1,1)
sarima(log(varve), 1, 1, 1, no.constant=TRUE)   # ARIMA(1,1,1)
       Estimate      SE   t.value p.value
  ar1    0.2330  0.0518    4.4994       0
  ma1   -0.8858  0.0292  -30.3861       0
  sigma^2 estimated as 0.2284
```

Hence the additional AR term is significant. The Q-statistic p-values for this model are also displayed in Figure 5.7, and it appears this model fits the data well.

As previously stated, the diagnostics are byproducts of the individual `sarima` runs. We note that we did not fit a constant in either model because there is no apparent drift in the differenced, logged varve series. This fact can be verified by noting the constant is not significant when the command `no.constant=TRUE` is removed in the code. ◊

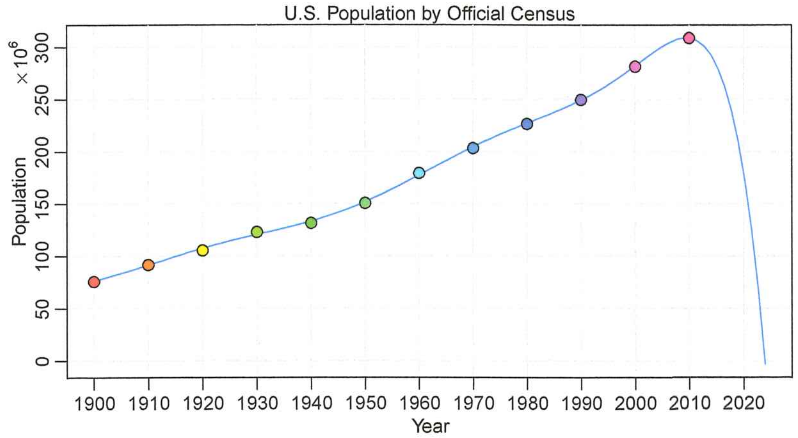


Figure 5.8 *A near perfect fit and a terrible forecast.*

In Example 5.6, we have two competing models, an AR(1) and an MA(2) on the GNP growth rate, that each appear to fit the data well. In addition, we might also consider that an AR(2) or an MA(3) might do better for forecasting. Perhaps combining both models, that is, fitting an ARMA(1, 2) to the GNP growth rate, would be the best. As previously mentioned, we have to be concerned with *overfitting* the model; it is not always the case that more is better. Overfitting leads to less-precise estimators, and adding more parameters may fit the data better but may also lead to bad forecasts. This result is illustrated in the following example.

Example 5.9. A Near Perfect Fit and a Terrible Forecast
Figure 5.8 shows the U.S. population by official census, every ten years from 1900 to 2010, as points. If we use these observations to predict the future population, we can fit a high degree polynomial so that the fit will be nearly perfect. There are twelve observations, so we could use an eight-degree polynomial to get a near perfect fit. The model in this case is

$$x_t = \beta_0 + \beta_1 t + \beta_2 t^2 + \cdots + \beta_8 t^8 + w_t.$$

The fitted line is also plotted in Figure 5.8 and it nearly passes through all the observations ($R^2 = 99.97\%$). The model predicts that the population of the United States will cross zero before 2025! This may or may not be true.

The R code to reproduce these results is as follows. We note that the data are not in `astsa` and there is a different R data set called `uspop`.

```
uspop = c(75.995, 91.972, 105.711, 123.203, 131.669,150.697,
              179.323, 203.212, 226.505, 249.633, 281.422, 308.745)
uspop = ts(uspop, start=1900, freq=.1)
t = time(uspop) - 1955
reg = lm( uspop~ t+I(t^2)+I(t^3)+I(t^4)+I(t^5)+I(t^6)+I(t^7)+I(t^8) )
   Multiple R-squared:  0.9997
```



```
b = as.vector(reg$coef)
g = function(t){ b[1] + b[2]*(t-1955) + b[3]*(t-1955)^2 +
                 b[4]*(t-1955)^3 + b[5]*(t-1955)^4 + b[6]*(t-1955)^5 +
                 b[7]*(t-1955)^6 + b[8]*(t-1955)^7 + b[9]*(t-1955)^8
}
par(mar=c(2,2.5,.5,0)+.5, mgp=c(1.6,.6,0))
curve(g, 1900, 2024, ylab="Population", xlab="Year", main="U.S.
                 Population by Official Census", panel.first=Grid(),
                 cex.main=1, font.main=1, col=4)
abline(v=seq(1910,2020,by=20), lty=1, col=gray(.9))
points(time(uspop), uspop, pch=21, bg=rainbow(12), cex=1.25)
mtext(expression(""%*% 10^6), side=2, line=1.5, adj=.95)
axis(1, seq(1910,2020,by=20), labels=TRUE)
```

◊

The final step of model fitting is model choice or model selection. That is, we must decide which model we will retain for forecasting. The most popular techniques, AIC, AICc, and BIC, were described in Section 3.1 in the context of regression models.

Example 5.10. Model Choice for the U.S. GNP Series
To follow up on Example 5.7, recall that two models, an AR(1) and an MA(2), fit the GNP growth rate well. In addition, recall that it was shown that the two models are nearly the same and are not in contradiction. To choose the final model, we compare the AIC, the AICc, and the BIC for both models. These values are a byproduct of the `sarima` runs.

```
sarima(diff(log(gnp)), 1, 0, 0)   # AR(1)
  $AIC: -6.456    $AICc: -6.456    $BIC: -6.425
sarima(diff(log(gnp)), 0, 0, 2)   # MA(2)
  $AIC: -6.459    $AICc: -6.459    $BIC: -6.413
```

The AIC and AICc both prefer the MA(2) fit, whereas the BIC prefers the simpler AR(1) model. The methods often agree, but when they do not, the BIC will select a model of smaller order than the AIC or AICc because its penalty is much larger. Ignoring the philosophical considerations that cause nerds to verbally assault each other, it seems reasonable to retain the AR(1) because pure autoregressive models are easier to work with. ◊

5.3 Seasonal ARIMA Models

In this section, we introduce several modifications made to the ARIMA model to account for seasonal behavior. Often, the dependence on the past tends to occur most strongly at multiples of some underlying seasonal lag s. For example, with monthly economic data, there is a strong yearly component occurring at lags that are multiples of $s = 12$, because of the strong connections of all activity to the calendar year. Data taken quarterly will exhibit the yearly repetitive period at $s = 4$ quarters. Natural phenomena such as temperature also have strong components corresponding to seasons. Hence, the natural variability of many physical, biological, and economic