Paul Shaman

The Impact of Model Fitting on the Spectrum

This course has emphasized that the aim in fitting a linear model to a (stationary) time series is to reduce the structure to that of white noise. This corresponds to modification of a colored spectral density, so that the modified density is flat (white). The operation which performs the modification specifies the model fit to the data.

It is instructive to examine how the fitting of an ARIMA model alters spectral structure. I will use ARIMA models fit to the Iowa and Australian beer data as illustrations.

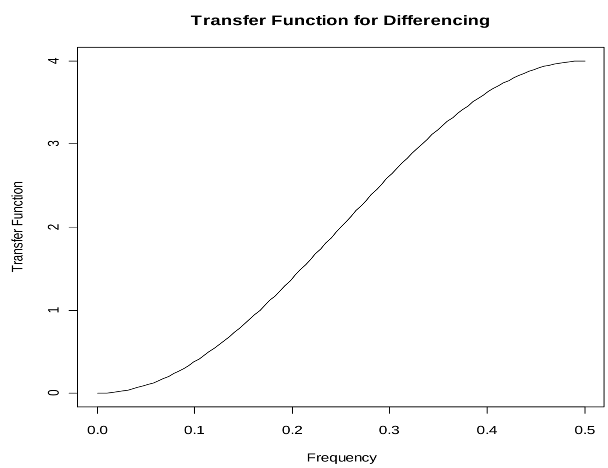
First, however, let's examine the operations which perform differencing and seasonal differencing. For a given time series y_t the differenced series is

$$(1 - B)\, y_t = y_t - y_{t-1}.$$

If the spectral density of y_t is $g(f)$, then the spectral density of the differenced series is

$$2(1 - \cos 2\pi f)\, g(f), \qquad -1/2 \le f \le 1/2.$$

The factor $2(1 - \cos 2\pi f)$ is the *transfer function* corresponding to the differencing operation. It alters the spectral density of y_t by multiplication. Here is the plot of this transfer function:



That is, the differencing operation attenuates low frequency activity and enhances high frequency activity in the spectral density of y_t.

Next let's examine seasonal differencing for period 12 (which is the seasonal period for the beer data). For a given time series y_t the seasonally differenced series is

$$(1 - B^{12})\, y_t \;=\; y_t - y_{t-12}.$$

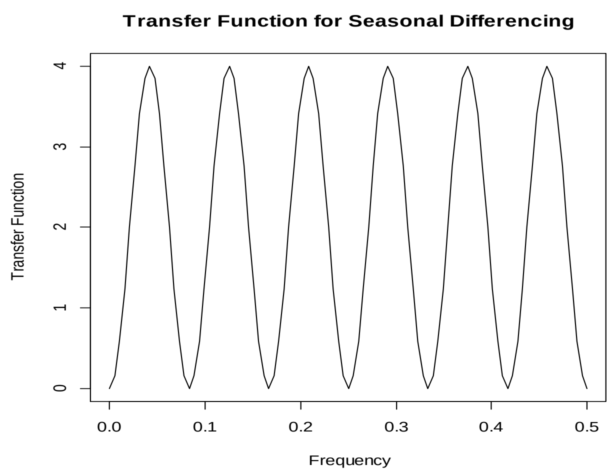
If the spectral density of y_t is $g(f)$, then the spectral density of the differenced series is

$$2(1 - \cos 2(12)\pi f)\, g(f), \qquad -1/2 \le f \le 1/2.$$

To fully understand the transfer function $2(1 - \cos 2(12)\pi f)$ from the seasonal differencing operation, we write $1 - B^{12}$ as a product of factors:

$$\begin{aligned}1 - B^{12} = {} & (1 - B)(1 - 2\cos(2\pi/12)B + B^2)(1 - 2\cos(4\pi/12)B + B^2) \\ & \cdot(1 - 2\cos(6\pi/12)B + B^2)(1 - 2\cos(8\pi/12)B + B^2) \\ & \cdot(1 - 2\cos(10\pi/12)B + B^2)(1 + B).\end{aligned}$$

The transfer function corresponding to this seasonal differencing operator attenuates the spectral density at frequencies 0, 1/12, 2/12, 3/12, 4/12, 5/12, and 6/12 (and alters and enhances the density in between these frequencies). The frequency 1/12 is the fundamental and the next five frequencies are its harmonics. Here is the plot of the transfer function:



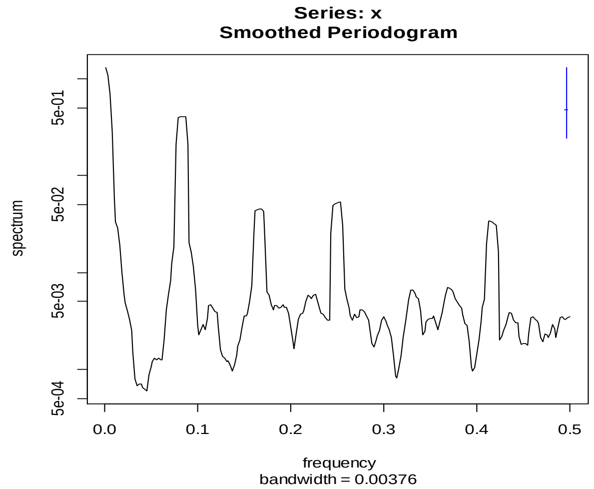
Let's examine the transfer function for the ARIMA model fit to the Australian beer data in the 13 April notes. At first we fit a regression to adjust the outlier at observation 317 and to remove the calendar components at frequencies 0.348 and 0.432. Then the following airline ARIMA model was fit to the regression residuals:

$$(1 - B)\left(1 - B^{12}\right) y_t = (1 - 0.8814B)\left(1 - 0.8270B^{12}\right)\varepsilon_t$$

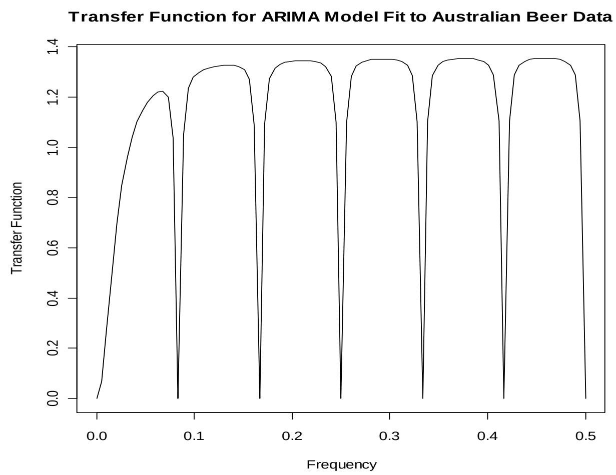
The transfer function corresponding to this fitted ARIMA model is

(1) $$\frac{4\left(1-\cos(2\pi f)\right)\left(1-\cos(24\pi f)\right)}{\left(1.7769-1.7628\cos(2\pi f)\right)\left(1.6839-1.6540\cos(24\pi f)\right)}.$$

The following spectral plot is that of the regression residuals.



This is the spectrum we want to flatten by building an ARIMA model with an appropriate corresponding transfer function. The spectral peak near frequency 0 indicates the presence of trend in the regression residuals. There are also peaks at frequencies 1/12, 1/6, 1/4, 1/3, and 5/12, all for seasonal structure. If the ARIMA model is to be successful in reducing the input to white noise, its associated transfer function will need to dampen these peaks and raise the valleys between the peaks. Here is the plot of the transfer function (1) corresponding to the fitted ARIMA model:



The peaks and valleys of the transfer function are properly positioned to reshape the residual spectrum so it becomes much more flat.

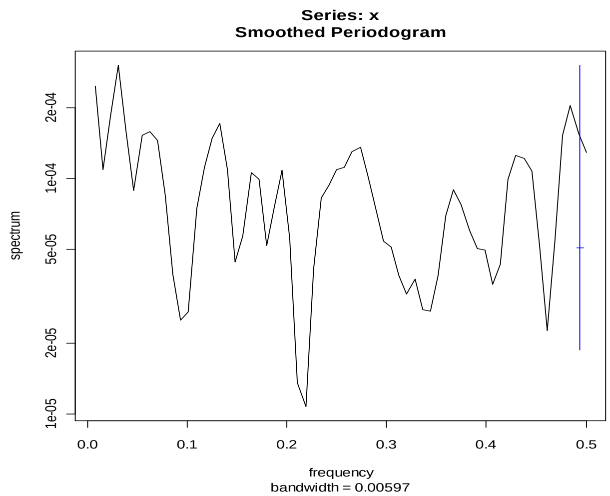
Let's also consider the transfer function corresponding to the first model fit to the Iowa data. The model is constructed in two steps, given on pages 3 and 5 of the 13 April notes. It is

$$(1 - 0.2990B^{16})(1 - 0.1535B - 0.1982B^2)\, y_t \;=\; 0.0084 + (1 + 0.2256B^4 + 0.1572B^8)\,\eta_t.$$

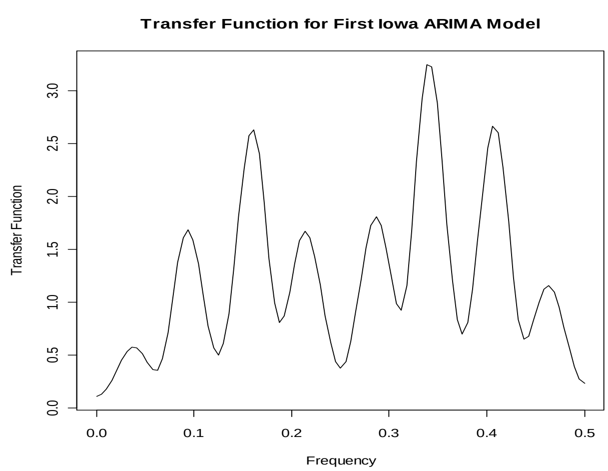
The transfer function corresponding to this model is

$$(2) \qquad \frac{\left(1.0894 - 0.5980\cos(32\pi f)\right)\left(1.0628 - 0.2462\cos(2\pi f) - 0.3964\cos(4\pi f)\right)}{1.0756 + 0.5221\cos(8\pi f) + 0.3144\cos(16\pi f)}$$

The following spectral plot is for the Iowa percentage change series.



And here is a plot of the transfer function (2):



The fitted model essentially has two components. There is the annual cyclical component, and for this spectral power is attenuated at frequencies ¼ and ½. The second component is the four-year cycle, and for this the spectral power is lowered at frequencies 1/16, 2/16, 3/16, 4/16, 5/16, 6/16, 7/16, and 8/16. The transfer function has the shape needed to flatten the spectral density of the percentage change time series.

Forecasting with ARMA Models

Suppose we have observed a time series $y_1, \ldots, y_T$ and we wish to forecast y_{T+h}, for some $h > 0$. We denote the minimum mean square error forecast by $y_{T+h,\,T}$. We are forecasting h steps ahead, given data up to and including time T.

Minimum mean square error forecasts are conditional expectations. The conditional expectations are calculated given observations up to and including time T. Within the context of an ARIMA model they are constructed using the actual values y_t for data already observed, and using conditional expectations for future values of y_t. For the white noise process ε_t one uses the realized values for the time points up to and including time T (the realized values are obtained as one-step-ahead forecast errors, what we have called residuals), and 0 (the conditional expectation) for future values of ε_t.

We calculate the minimum mean square error forecasts assuming that all parameter values of the process are known. In practice one then substitutes estimates for the unknown parameter values in the forecast expressions. So, in the discussion below, we develop forecasts assuming all parameter values are known.

To illustrate the procedure consider an ARMA(2,2) model with mean μ,

$$(y_t - \mu) - \varphi_1(y_{t-1} - \mu) - \varphi_2(y_{t-2} - \mu) = \varepsilon_t + \theta_1\varepsilon_{t-1} + \theta_2\varepsilon_{t-2} \,.$$

Recall we have data up to and including time T. We use the ARMA equation to construct the forecast. Write

$$y_{T+h} = \mu + \varphi_1(y_{T+h-1} - \mu) + \varphi_2(y_{T+h-2} - \mu) + \varepsilon_{T+h} + \theta_1\varepsilon_{T+h-1} + \theta_2\varepsilon_{T+h-2} \,.$$

For h=1 we apply the conditional expectation operation to each side and obtain

$$y_{T+1,T} = \mu + \varphi_1(y_T - \mu) + \varphi_2(y_{T-1} - \mu) + \theta_1\varepsilon_T + \theta_2\varepsilon_{T-1} \,.$$

For h=2 we then have

$$y_{T+2,\mathrm{T}} = \mu + \varphi_1(y_{T+1,T} - \mu) + \varphi_2(y_T - \mu) + \theta_2\varepsilon_T \,.$$

The other forecasts are

$$y_{T+h,T} = \mu + \varphi_1(y_{T+h-1,T} - \mu) + \varphi_2(y_{T+h-2,T} - \mu), \qquad h=3, 4, \ldots \,.$$

This scheme requires calculation for h=1 first, followed by h=2, and so on.

There is an alternative way to express the forecast. Consider the ARMA(p,q) model,

$$(1-\phi_1 B-\cdots-\phi_p B^p)(y_t-\mu)=(1+\theta_1 B+\cdots+\theta_q B^q)\,\varepsilon_t\,.$$

Rewrite this as an infinite order moving average,

$$y_t=\mu+(1-\phi_1 B-\cdots-\phi_p B^p)^{-1}(1+\theta_1 B+\cdots+\theta_q B^q)\varepsilon_t$$

$$=\mu+(1+\psi_1 B+\psi_2 B^2+\psi_3 B^3+\cdots)\varepsilon_t$$

$$=\mu+\varepsilon_t+\psi_1\varepsilon_{t-1}+\psi_2\varepsilon_{t-2}+\psi_3\varepsilon_{t-3}+\cdots.$$

Then

$$y_{T+h}=\mu+\varepsilon_{T+h}+\psi_1\varepsilon_{T+h-1}+\psi_2\varepsilon_{T+h-2}+\cdots,$$

and the h-step-ahead forecast is thus

$$y_{T+h,T}=\mu+\psi_h\varepsilon_T+\psi_{h+1}\varepsilon_{T-1}+\psi_{h+2}\varepsilon_{T-2}+\cdots.$$

Therefore the forecast error is

$$e_{T+h,T}=y_{T+h}-y_{T+h,T}$$

$$=\varepsilon_{T+h}+\psi_1\varepsilon_{T+h-1}+\cdots+\psi_{h-1}\varepsilon_{T+1}\,.$$

The variance of this forecast error is

$$Var(e_{T+h,T})=\sigma^2(1+\psi_1^2+\cdots+\psi_{h-1}^2),$$

where σ^2 is the variance of ε_t. For an approximate 95 per cent prediction interval for the forecast of y_{T+h} we use

$$y_{T+h,T}\pm 1.96\sigma(1+\psi_1^2+\cdots+\psi_{h-1}^2)^{1/2}.$$

In practice parameter estimates are used in place of the exact parameter values, and this is assumed in the discussion which follows.

Let’s look at some examples.

1. AR(1). The model is $(y_t - \mu) - \varphi_1(y_{t-1} - \mu) = \varepsilon_t$. Then

$$y_{T+1,T} = \mu + \varphi_1(y_T - \mu),$$

$$y_{T+2,T} = \mu + \varphi_1(y_{T+1,T} - \mu) = \mu + \varphi_1^2(y_T - \mu),$$

$$y_{T+3,T} = \mu + \varphi_1(y_{T+2,T} - \mu) = \mu + \varphi_1^3(y_T - \mu),$$

.

.

.

$$y_{T+h,T} = \mu + \varphi_1(y_{T+h-1,T} - \mu) = \mu + \varphi_1^h(y_T - \mu), \quad h = 1, 2, \ldots.$$

Comments:

- Since the autoregressive parameter satisfies $-1 < \varphi_1 < 1$, the forecasts tend to the mean, μ, as the forecast horizon increases. The closer the parameter value is to zero, the more rapid is this convergence to the mean level.
- For this AR(1) model, the forecasts depend directly on the data only through the last data value (apart from the fact that the parameters are estimated from all of the data, of course).

Let's consider the data set nyseadv. Recall that an AR(1) can be fit to the data. Here are the model estimation and the forecasts up to a horizon of 12.

```
> advances<-read.csv("G:/Stat71122Spring/nyseadv.txt")
> attach(advances)
> head(advances)
  pctadvnc
1    27.22
2    37.85
3    29.20
4    42.77
5    34.81
6    14.14
> advances.ts<-ts(advances)
> advar1<-arima(advances.ts,order=c(1,0,0))
> advar1

Call:
arima(x = advances.ts, order = c(1, 0, 0))
```

```
Coefficients:
         ar1  intercept
      0.2106    38.5796
s.e.  0.0800     1.1421

sigma^2 estimated as 121.5:  log likelihood = -569.05,  aic = 1144.11
```

The forecasts follow. Prediction errors are also given. Note the rapid convergence to the estimate of the mean (intercept).

```
> predict(advar1,n.ahead=12)
$pred
Time Series:
Start = 150
End = 161
Frequency = 1

 [1] 38.49751 38.56228 38.57593 38.57880 38.57940 38.57953 38.57956 38.57956
 [9] 38.57957 38.57957 38.57957 38.57957

$se
Time Series:
Start = 150
End = 161
Frequency = 1

 [1] 11.02382 11.26573 11.27634 11.27681 11.27683 11.27683 11.27683 11.27683
 [9] 11.27683 11.27683 11.27683 11.27683
```

The last data value is 38.19. The prediction for $h=1$ is

$$38.579562 + 0.210642(38.19 - 38.579562) = 38.497504\ .$$

2. AR(p). The model is $(y_t - \mu) - \varphi_1(y_{t-1} - \mu) - \cdots - \varphi_1(y_{t-p} - \mu) = \varepsilon_t$. The forecasts behave similarly to those for the AR(1) model.

Comments:

- We assume that the zeros of the autoregressive polynomial written in the form

$$z^p - \varphi_1 z^{p-1} - \varphi_2 z^{p-2} - \cdots - \varphi_p$$

  are strictly less than one in absolute value, as per usual. It follows that the forecasts tend to the mean, μ, as the forecast horizon increases. The closer the zeros of the polynomial are to zero, the more rapid is this convergence to the mean level.

- For the AR(p) model, the forecasts depend on the data only through the last p data values (and the parameters are estimated from all of the data, of course).

As an example, let's look at the Iowa data and fit an AR(16) model. This is overkill, but it does reduce the data to white noise. Here is the table of parameter estimates:

```
> coeftest(model)

z test of coefficients:

            Estimate Std. Error z value  Pr(>|z|)
ar1        0.1165675  0.0863402  1.3501   0.17699
ar2        0.1776014  0.0878461  2.0217   0.04320 *
ar3       -0.0129788  0.0887750 -0.1462   0.88376
ar4        0.1814916  0.0884543  2.0518   0.04019 *
ar5        0.0245707  0.0904024  0.2718   0.78578
ar6        0.0448060  0.0924030  0.4849   0.62775
ar7        0.0796877  0.0913070  0.8727   0.38280
ar8        0.1115613  0.0923543  1.2080   0.22706
ar9       -0.1438879  0.0962305 -1.4952   0.13485
ar10      -0.0208799  0.0959483 -0.2176   0.82773
ar11       0.0944558  0.0956839  0.9872   0.32356
ar12      -0.0442884  0.0983356 -0.4504   0.65244
ar13      -0.0980121  0.0973301 -1.0070   0.31393
ar14       0.0382715  0.0980739  0.3902   0.69637
ar15       0.0443614  0.0954548  0.4647   0.64212
ar16       0.2253596  0.0962300  2.3419   0.01919 *
intercept  0.0191343  0.0036655  5.2201 1.788e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

For this model fit the forecasts converge to the estimate of the mean, 0.0191343, but very slowly. There is slow convergence because some of the zeros of the autoregressive polynomial have magnitude close to 1. Here are some of the forecasts:

```
> predict(model,n.ahead=300)
$pred
Time Series:
Start = 129
End = 428
Frequency = 1

  [1] 0.02293988 0.02529377 0.02383661 0.02516356 0.02257091 0.02776700
  [7] 0.02444068 0.02030917 0.02252780 0.02571087 0.02424472 0.02427587

[103] 0.01949437 0.01949229 0.01947142 0.01947625 0.01945695 0.01945585
[109] 0.01943776 0.01943999 0.01942270 0.01942147 0.01940707 0.01940801

[289] 0.01913644 0.01913639 0.01913633 0.01913627 0.01913622 0.01913616
[295] 0.01913611 0.01913606 0.01913601 0.01913597 0.01913592 0.01913587
```

Convergence has not been established even at 300 time points ahead. Although an AR(4) model is not an adequate fit to the data, let's also consider its forecasts.

```
> coeftest(model2)

z test of coefficients:

           Estimate Std. Error z value  Pr(>|z|)
ar1       0.1459119  0.0855766  1.7050  0.088186 .
ar2       0.1952756  0.0867310  2.2515  0.024353 *
ar3       0.0155083  0.0875213  0.1772  0.859355
ar4       0.2617809  0.0861843  3.0375  0.002386 **
intercept 0.0184134  0.0021488  8.5693 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> predict(model2,n.ahead=80)
$pred
Time Series:
Start = 129
End = 208
Frequency = 1

 [1] 0.02102102 0.02283703 0.02293073 0.02131025 0.02046943 0.02050716
 [7] 0.02034787 0.01989474 0.01957799 0.01945070 0.01932154 0.01915431

[49] 0.01841458 0.01841439 0.01841423 0.01841410 0.01841398 0.01841389
[55] 0.01841381 0.01841374 0.01841368 0.01841363 0.01841359 0.01841356

[73] 0.01841340 0.01841340 0.01841340 0.01841339 0.01841339 0.01841339
[79] 0.01841339 0.01841339
```

Convergence for this model is more rapid.

3. MA(1). The model is $y_t = \mu + \varepsilon_t + \theta_1 \varepsilon_{t-1}$. This implies

$$y_{T+1,T} = \mu + \theta_1 \varepsilon_T ,$$

$$y_{T+h,T} = \mu, \qquad h = 2, 3, \ldots .$$

Thus, the first forecast is a deviation from the mean level, and it uses the data only via the last residual value (apart from the fact that the parameters are estimated from all of the data, of course). Moreover, all subsequent forecasts are equal to the mean level.

4. MA(q). The first q forecasts are deviations from the mean level, and they use the data only via the last q residual values (apart from the fact that the parameters are estimated from all of the data, of course). All forecasts with horizon greater than q are equal to the mean level.

Let's consider an MA(5) fit to the quarterly U.S. gnp return data, in the file qgnp. It's a decent fit to the data.

```
> coeftest(qgnpma5)

z test of coefficients:

            Estimate Std. Error z value  Pr(>|z|)
ma1        0.3219571  0.0742250  4.3376 1.441e-05 ***
ma2        0.3349962  0.0812659  4.1222 3.752e-05 ***
ma3        0.0829112  0.0855773  0.9688   0.33262
ma4       -0.0806020  0.0841519 -0.9578   0.33816
ma5       -0.1436905  0.0785182 -1.8300   0.06725 .
intercept  0.0077056  0.0010983  7.0160 2.283e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> predict(qgnpma5,n.ahead=10)
$pred
Time Series:
Start = 177
End = 186

Frequency = 1

 [1] 0.001130748 0.004357913 0.008133897 0.009978086 0.009171849 0.007705582
 [7] 0.007705582 0.007705582 0.007705582 0.007705582
```

5. ARMA(p, q). As the horizon increases, the forecasts converge to the mean level. All of the forecasts depend upon the last p data points and the last q residual values (apart from the fact that the parameters are estimated from all of the data, of course). See the discussion of the ARMA(2, 2) model on page 1.

6. Models with a seasonal component. The previous examples describe how to analyze these models. For example, suppose one has fit an ARMA(p, q)$(P,Q)_s$ model. We may view this as an ARMA($p + Ps$, $q + Qs$) model (the model will have a lot of zero coefficients, of course).

7. Models with differencing. Derivation of the properties of these models is more complicated than that for models without differencing. Let's look in detail at a simple special case, an ARIMA(1,1,0) model. It is written as

$$(1 - \varphi_1 B)((1 - B)\, y_t - \mu) = \varepsilon_t \,.$$

Tedious algebraic calculation shows that

$$y_{T+h,T} = \frac{1-\varphi_1^{h+1}}{1-\varphi_1} y_T - \frac{\varphi_1(1-\varphi_1^h)}{1-\varphi_1} y_{T-1} + \frac{h(1-\varphi_1) - \varphi_1(1-\varphi_1^h)}{1-\varphi_1} \mu \,. \tag{1}$$

For large h the right-hand side of (1) is approximately equal to

(2) $$\frac{1}{1-\varphi_1}(y_T - \varphi_1 y_{T-1}) + h\mu - \frac{\varphi_1}{1-\varphi_1}\mu.$$

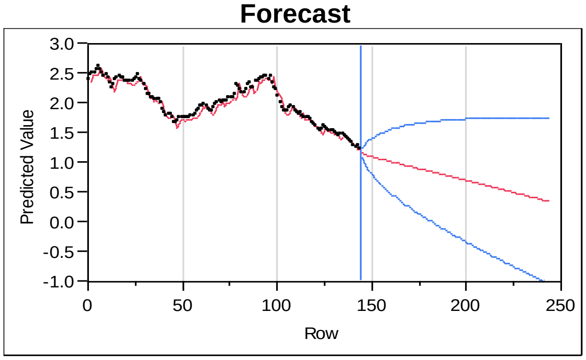
Recall the file exchange.jmp, which gives the U.S. dollar-U.K. sterling exchange rate, in dollars per pound, for 144 consecutive months. An ARIMA(1,1,0) model is a good fit to these data.

Proper forecasting for an ARIMA model which includes differencing (d = 1) requires estimation of the mean *after differencing*. The `arima` command we have been using does not allow mean estimation when there is differencing. To illustrate forecasting in the present example, I'll use the results from JMP software. The "forecast" package in R does give mean estimation in this case.

Parameter Estimates

Term	Lag	Estimate	Std Error	t Ratio	Prob>|t|	Constant Estimate
AR1	1	0.4125745	0.0774670	5.33	<.0001	-0.0046953
Intercept	0	-0.0079930	0.0059003	-1.35	0.1777	

Forecast



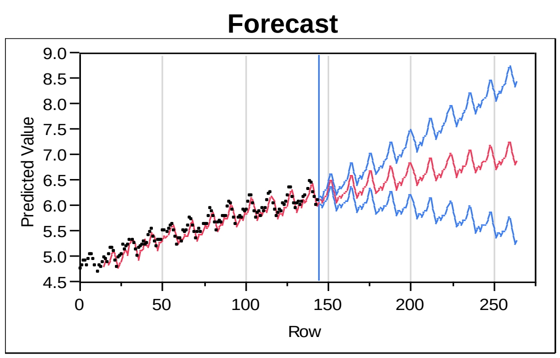
Here are the forecasts:

h	Forecast
1	1.15520053
2	1.13903589
3	1.12767148
4	1.11828751
5	1.10972063
6	1.10149086
7	1.09340018
8	1.08536687
9	1.07735724
10	1.06935737
11	1.06136154
12	1.05336736
13	1.04537388

14	1.03738067
15	1.02938759
16	1.02139455
17	1.01340153
18	1.00540852
19	0.99741551
20	0.98942251

Note that $y_{T+20,T} - y_{T+19,T} = -0.0079930 = \hat{\mu}$, in agreement with formula (2) on page 14. Also, for this data set y_T = 1.183 and y_{T-1} = 1.239, and it is easy to verify that the numbers in the table above agree with formula (1) on page 12.

As this example illustrates, the forecasts for an ARIMA model with first-order regular differencing converge to a straight line trend as the horizon increases. The previous examples showed that ARMA (no differencing) forecasts converge to a constant level as the horizon increases. What is the structure of the forecasts when there is seasonal differencing? In this case the forecasts converge to a repeated seasonal pattern, and this seasonal pattern will fluctuate about a constant level if there is no regular differencing and will fluctuate about a trend line if there is regular differencing. As an example, consider the airline model ARIMA(0,1,1)(0,1,1)$_{12}$ fit to the logged international airline data.



How do you think the forecasts will behave if there is regular differencing of order 2?

A comment: Typically forecasting is problematic. And given the structures of forecast curves, as illustrated above, it is inadvisable to attempt to forecast very far into the future. Forecasts are likely to be reasonably accurate as long as there are no abrupt changes in the direction of the time series, relative to what has already transpired. However, such changes often do occur, and we must conclude that our ability to construct accurate forecasts is very often rather limited.