Homework 7

Benjamin Noland

1. (Cryer & Chan, Exercise 9.1)

   a. We have

   $$\begin{aligned}\hat{Y}_t(1) - \mu &= \mathrm{E}(Y_{t+1} - \mu | Y_1, \ldots, Y_t) \\ &= \mathrm{E}[\phi(Y_t - \mu) + e_{t+1} | Y_1, \ldots, Y_t] \\ &= \phi[\mathrm{E}(Y_t | Y_1, \ldots, Y_t) - \mu] + \mathrm{E}(e_{t+1} | Y_1, \ldots, Y_t) \\ &= \phi(Y_t - \mu) + \mathrm{E}(e_{t+1}) \\ &= \phi(Y_t - \mu),\end{aligned}$$

   so that

   $$\hat{Y}_t(1) = \phi(Y_t - \mu) + \mu.$$

   Plugging in the given estimates for Y_t, ϕ, and μ, we get $\hat{Y}_t(1) = 10.1$.

   b. (**In only one way**) Let $l > 1$. Then we have

   $$\begin{aligned}\hat{Y}_t(l) - \mu &= \mathrm{E}(Y_{t+l} - \mu | Y_1, \ldots, Y_t) \\ &= \mathrm{E}[\phi(Y_{t+l-1} - \mu) + e_{t+l} | Y_1, \ldots, Y_t] \\ &= \phi[\mathrm{E}(Y_{t+l-1} | Y_1, \ldots, Y_t) - \mu] + \mathrm{E}(e_{t+l} | Y_1, \ldots, Y_t) \\ &= \phi(\hat{Y}_t(l) - \mu) + \mathrm{E}(e_{t+l}) \\ &= \phi(\hat{Y}_t(l) - \mu) \\ &= \phi^l(Y_t - \mu),\end{aligned}$$

   where in the last step the recursion in l was evaluated using the result from part (a). Therefore,

   $$\hat{Y}_t(l) = \phi^l(Y_t - \mu) + \mu.$$

   In particular,

   $$\hat{Y}_t(2) = \phi^2(Y_t - \mu) + \mu.$$

   Plugging in the given estimates for Y_t, ϕ, and μ, we get $\hat{Y}_t(2) = 11.15$.

   c. Using the result from part (b), we get

   $$\hat{Y}_t(10) = \phi^{10}(Y_t - \mu) + \mu.$$

   Plugging in the given estimates for Y_t, ϕ, and μ, we get $\hat{Y}_t(10) \approx 10.8$.

2. (Cryer & Chan, Exercise 9.2)

   a. The model is of the form

   $$Y_t = \theta_0 + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + e_t.$$

The forecast at lead l is given by

$$\begin{aligned}
\hat{Y}_t(l) &= \mathrm{E}(Y_{t+l}|Y_1,\ldots,Y_t) \\
&= \mathrm{E}(\theta_0 + \phi_1 Y_{t+l-1} + \phi_2 Y_{t+l-2} + e_{t+l}|Y_1,\ldots,Y_t) \\
&= \theta_0 + \phi_1 \hat{Y}_t(l-1) + \phi_2 \hat{Y}_t(l-2) + \mathrm{E}(e_{t+l}) \\
&= \theta_0 + \phi_1 \hat{Y}_t(l-1) + \phi_2 \hat{Y}_t(l-2).
\end{aligned}$$

Thus, in particular,

$$\begin{aligned}
\hat{Y}_t(1) &= \theta_0 + \phi_1 \hat{Y}_t(0) + \phi_2 \hat{Y}_t(-1) \\
&= \theta_0 + \phi_1 Y_t + \phi_2 Y_{t-1}
\end{aligned}$$

and

$$\begin{aligned}
\hat{Y}_t(2) &= \theta_0 + \phi_1 \hat{Y}_t(1) + \phi_2 \hat{Y}_t(0) \\
&= \theta_0 + \phi_1 \hat{Y}_t(1) + \phi_2 Y_t.
\end{aligned}$$

We are given that $Y_1 = 9, Y_2 = 11$, and $Y_3 = 10$ (in millions of dollars). Plugging these values into the above expressions, with $t = 3$, we find $\hat{Y}_3(1) = 10.5$ and $\hat{Y}_3(2) = 11.55$ for the forecasted sales for 2008 and 2009, respectively.

b. Using the result on page 75, we find that $\Psi_1 = \phi_1 = 1.1$.

c. Assume the white noise process $\{e_t\}$ is Gaussian. The forecast error at lead 1 is given by (see the result on page 202)

$$\begin{aligned}
e_t(1) &= e_{t+1} + \Psi_1 e_t + \Psi_0 e_{t+1} \\
&= e_{t+1} + \phi_1 e_t + e_{t+1} \\
&= \phi_1 e_t + 2e_{t+1}
\end{aligned}$$

since $\Psi_1 = \psi_1$ and $\Psi_0 = 1$ (see the result on page 75). Thus,

$$e_t(1) = \phi_1 e_t + 2e_{t+1} \sim \mathrm{N}(0, (\phi_1^2 + 4)\sigma_e^2).$$

Since $e_t(1) = Y_{t+1} - Y_t(1)$ by definition, we therefore get

$$\mathrm{P}\left\{ -z_{1-\alpha/2} \leq \frac{Y_{t+1} - \hat{Y}_t(1)}{\sqrt{(\phi_1^2 + 4)\sigma_e^2}} \leq z_{1-\alpha/2} \right\} = 1 - \alpha$$

for a given confidence level α. Rearrangement yields the following $(1-\alpha) \times 100\%$ prediction interval for Y_{t+1}:

$$\hat{Y}_t(1) \pm z_{1-\alpha/2}\sqrt{(\phi_1^2 + 4)\sigma_e^2}.$$

Letting $\alpha = 0.05$ and plugging in the pertinent quantities, we get $(4.14, 16.8)$ as a 95% prediction interval for Y_4.

3. The following simulates an MA(2) process with $\theta_1 = 1, \theta_2 = -0.6$, and $\mu = 100$:

```r
set.seed(1432756)
series <- arima.sim(model = list(ma = c(-1, 0.6)), n = 36) + 100
```

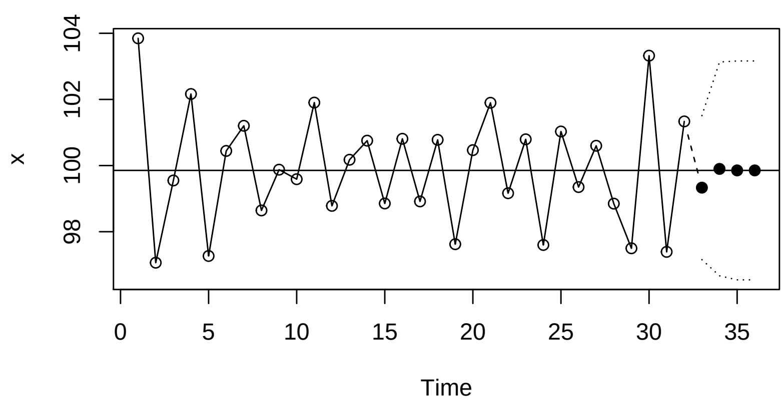
a. The following uses the first 32 observations of the process to compute maximum likelihood estimates of θ_1, θ_2, and μ:

```r
# Set aside the first 32 values for computing forecasts.
training <- window(series, end = 32)
model <- arima(training, order = c(0, 0, 2), method = "ML")
model
```

```
##
## Call:
## arima(x = training, order = c(0, 0, 2), method = "ML")
##
## Coefficients:
##           ma1    ma2  intercept
##        -1.097  0.326      99.85
## s.e.    0.154  0.168       0.05
##
## sigma^2 estimated as 1.23:  log likelihood = -49.5,  aic = 105
```

b. The following plots the time series along with the 4 forecasted values (the solid black points), pointwise 95% prediction limits, and a solid horizontal line indicating the estimated value of μ:

```r
res <- plot(model, n.ahead = 4, pch = 19)
abline(h = coef(model)[["intercept"]])
```



The following are the four predicted values:

```r
preds <- res$pred
preds
```

```
## Time Series:
## Start = 33
## End = 36
```

```
## Frequency = 1
## [1] 99.3 99.9 99.9 99.9
```

c. The forecasts at lead times 3 and 4 are *very* close to the estimated value of μ, and therefore lie almost along the horizontal line in the plot in part (b).

d. The true values are as follows:

```r
truth <- window(series, start = 33)
truth
```

```
## Time Series:
## Start = 33
## End = 36
## Frequency = 1
## [1]  98.2 101.6  97.7 103.1
```

We therefore have the following pointwise deviations:

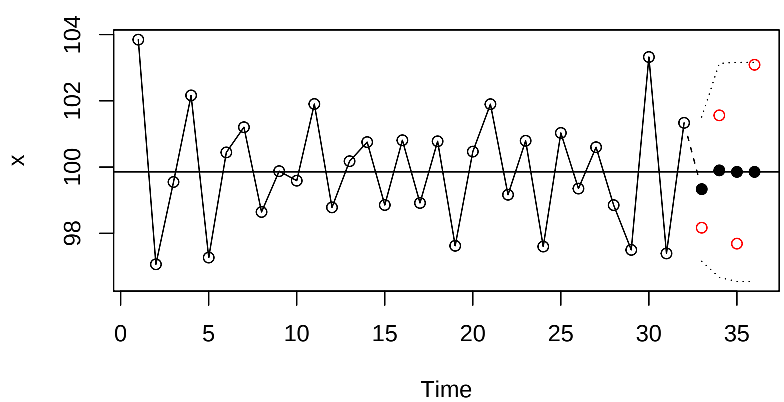
```r
as.numeric(preds) - as.numeric(truth)
```

```
## [1]  1.16 -1.66  2.17 -3.24
```

Thus the model overestimates the true values at lead times 1 and 3, and underestimates the true values at lead times 2 and 4.

e. The following is the same plot as in part (b), but with the true values plotted as red circles. Each of the actual values appears to lie within its corresponding 95% prediction interval.

```r
plot(model, n.ahead = 4, pch = 19)
points(y = truth, x = time(truth), col = 2)
abline(h = coef(model)[["intercept"]])
```



f. The following code simulates the same process as above 500 times and computes the fraction of times the forecast limits cover each of the true values.

```r
set.seed(1432756)
```

```r
N <- 500
count <- 0
for (i in 1:N) {
  # Simulate the process and fit a model on the training subset.
  series <- arima.sim(model = list(ma = c(-1, 0.6)), n = 36) + 100
  training <- window(series, end = 32)
  model <- arima(training, order = c(0, 0, 2), method = "ML")

  # Compute the forecasts and get the upper/lower prediction limits.
  res <- plot(model, n.ahead = 4, Plot = FALSE)
  upper <- res$upi
  lower <- res$lpi

  # Do all the forecasts lie within their respective prediction intervals?
  truth <- window(series, start = 33)
  success <- all(truth >= lower & truth <= upper)
  count <- count + success
}

count / N
```

```
## [1] 0.784
```

Thus the forecast limits cover each of the true values approximately 78.4% of the time.

4. (Cryer & Chan, Exercise 9.23)

   a. The following code fits an IMA(1, 1) model to all but the last five values of the time series:

```r
data(robot)

training <- window(robot, end = length(robot) - 5)
model <- arima(training, order = c(0, 1, 1))
model
```

```
##
## Call:
## arima(x = training, order = c(0, 1, 1))
##
## Coefficients:
##           ma1
##       -0.878
## s.e.   0.042
##
## sigma^2 estimated as 6.07e-06:  log likelihood = 1458,  aic = -2914
```

Next, we use this model to forecast the final five observations in the time series, which were left out when fitting the model:

```
res <- plot(model, n.ahead = 5, Plot = FALSE)
```

The forecasts are

```
as.numeric(res$pred)
```

```
## [1] 0.000769 0.000769 0.000769 0.000769 0.000769
```

and the upper and lower prediction limits are

```
as.numeric(res$upi)
```

```
## [1] 0.00560 0.00563 0.00567 0.00571 0.00574
```

and

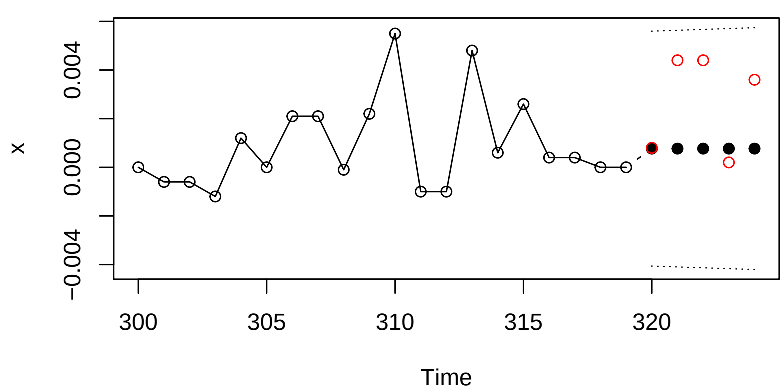
```
as.numeric(res$lpi)
```

```
## [1] -0.00406 -0.00410 -0.00413 -0.00417 -0.00420
```

respectively. Notice that each of the forecasts are the same, as should be the case for an IMA(1, 1) model with no constant term (as is the case here).

b. The following is a plot of the time series, along with the forecasts (solid black points) and their associated 95% prediction limits, as well as the associated true values (red circles). The plot is restricted to the end of the time series for easy visualization.

```
plot(model, n.ahead = 5, n1 = 300, pch = 19)
truth <- window(robot, start = length(robot) - 5 + 1)
points(y = truth, x = time(truth), col = 2)
```



From this plot we see that each of the true values lies within the prediction limits for the corresponding forecast. As mentioned in part (a), all of the forecasted values are the same, and this is shown on the plot.

c. The following code fits an ARMA(1, 1) model to all but the last five values of the time series:

```r
training <- window(robot, end = length(robot) - 5)
model <- arima(training, order = c(1, 0, 1))
model
```

```
##
## Call:
## arima(x = training, order = c(1, 0, 1))
##
## Coefficients:
##          ar1     ma1  intercept
##        0.948  -0.807      0.001
## s.e.  0.031   0.062      0.000
##
## sigma^2 estimated as 5.96e-06:  log likelihood = 1466,  aic = -2926
```

Next, we use this model to forecast the final five observations in the time series, which were left out when fitting the model:

```r
res <- plot(model, n.ahead = 5, Plot = FALSE)
```

The forecasts are

```r
as.numeric(res$pred)
```

```
## [1] 0.000942 0.000967 0.000990 0.001012 0.001033
```

and the upper and lower prediction limits are

```r
as.numeric(res$upi)
```

```
## [1] 0.00573 0.00580 0.00586 0.00592 0.00598
```

and

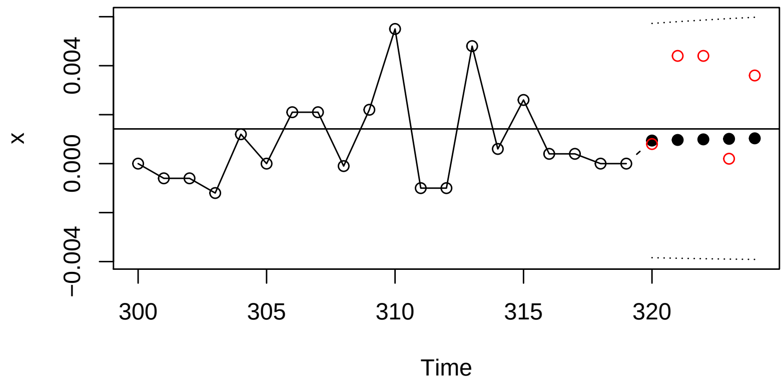
```r
as.numeric(res$lpi)
```

```
## [1] -0.00384 -0.00386 -0.00388 -0.00390 -0.00391
```

respectively. Note that unlike with the IMA(1, 1) model fit in part (a), the forecasts are not all the same value.

The following is a plot of the time series, along with the forecasts (solid black points) and their associated 95% prediction limits, as well as the associated true values (red circles). The constant term in the model is indicated by the solid horizontal line. The plot is restricted to the end of the time series for easy visualization.

```r
plot(model, n.ahead = 5, n1 = 300, pch = 19)
abline(h = coef(model)[["intercept"]])
truth <- window(robot, start = length(robot) - 5 + 1)
points(y = truth, x = time(truth), col = 2)
```



As mentioned above, unlike with the IMA(1, 1), the forecasts are not all the same value, and this can be seen in the plot. However, as with the IMA(1, 1) model, each of the true values lies within the prediction limits for the corresponding forecast.

5. Let $\tilde{\phi}(x) = (1 - 1.6x + 0.7x^2)(1 - 0.8x^{12})$ denote the AR characteristic polynomial in question.

   a. The AR characteristic polynomial $\tilde{\phi}$ has roots

$$\begin{aligned} x_1 &\approx 1.14 - 0.350i \\ x_2 &\approx 1.13 + 0.350i \\ x_3 &\approx -1.02 \\ x_4 &\approx 1.02, \end{aligned}$$

   each of which has modulus > 1, so that the process is stationary.

   b. The AR characteristic polynomial $\tilde{\phi}$ can be written as $\tilde{\phi}(x) = \phi(x)\Phi(x)$, where

$$\begin{aligned} \phi(x) &= 1 - 1.6x + 0.7x^2 \\ \Phi(x) &= 1 - 0.8x^{12}. \end{aligned}$$

   Since the model is AR, the MA characteristic polynomial of the model is simply $\tilde{\theta}(x) = 1$. Therefore $q = Q = 0$, $p = 2$, $P = 1$, and $s = 12$, so that the model is $\text{ARMA}(2, 0) \times (1, 0)_{12}$.