Demand function herring

Lars J. Ravn-Jonsen

June 27, 2006

1 Data

Data from Arnason *et al.* (2000) is updated with Fiskeridirektoratet (2006*a*,*b*) so the time series is now 1982–2005, i.e. 24 observations. Harvest in ton and value in 1.000 DKK. Price is calculate as value divided by landings, hence price is in 1.000 DKK pr. ton or DKK pr. kg. Nominal price is converted to real price with CPI (Danmarks Statistik, 2006) with base of 2004 and converted to NOK by exchange rate 100DKK=90.9300NOK (1/6 2004).

```
> setwd("F:\\Dokumenter\\feedback\\sild")
> library(gdata)
> a <- read.xls("pris_sild.xls", sheet = 3, perl = "c:/perl/bin/perl")
> a$price <- a$price * 0.9093

> a

   year landing    price
1  1982  81.000 3.864613
2  1983 172.000 2.204645
3  1984 124.000 2.778501
4  1985 136.000 2.969786
5  1986 150.000 2.248954
6  1987 157.000 1.816671
7  1988 184.000 1.835043
8  1989 171.000 2.018763
9  1990 136.000 2.183031
10 1991 146.000 2.140883
11 1992 156.000 2.175466
12 1993 169.000 1.918257
13 1994 178.000 1.853415
14 1995 191.000 1.463279
15 1996 153.009 1.356541
16 1997 125.302 1.524046
17 1998 139.711 1.558200
18 1999 137.578 1.298876
```

```
19 2000 153.899 1.129468
20 2001 141.508 2.220865
21 2002 112.582 2.457409
22 2003 114.806 1.729254
23 2004 136.809 1.521349
24 2005 167.450 1.881092
```

2 Model

A linear model is used to model the real price:

$$p_i = \alpha + \beta h_i + \epsilon_i$$

where p_i is average real price in NOK pr.kg. (or 1.000 NOK pr ton) of herring in Denmark in year i, h_i is the amount of herring in ton landed from Danish fishing vessels in year i and $i = 1982, 1983, \ldots, 2005$. This model yields residuals with high autocorrelation, hence the model is attempted corrected with autocorrelation of the AR(1), AR(2). This do however not yield god results and moving average is included in the modeling in the form of the ARMA(0,1), ARMA(1,1), ARMA(0,2), ARMA(1,2) and ARMA(0,3) type:

model (0,0) ϵ_i assumed NID(o, σ^2)

model (1,0) $\epsilon_i = \phi\epsilon_{i-1} + \nu_i$ where ν_i assumed NID(o, σ^2)

model (2,0) $\epsilon_i = \phi_1\epsilon_{i-1} + \phi_2\epsilon_{i-2} + \nu_i$ where ν_i assumed NID(o, σ^2)

model (0,1) $\epsilon_i = \theta\nu_{i-1} + \nu_i$ where ν_i assumed NID(o, σ^2)

model (1,1) $\epsilon_i = \phi\epsilon_{i-1} + \theta\nu_{i-1} + \nu_i$ where ν_i assumed NID(o, σ^2)

model (0,2) $\epsilon_i = \theta_1\nu_{i-1} + \theta_2\nu_{i-2} + \nu_i$ where ν_i assumed NID(o, σ^2)

model (1,2) $\epsilon_i = \phi\epsilon_{i-1} + \theta_1\nu_{i-1} + \theta_2\nu_{i-2} + \nu_i$ where ν_i assumed NID(o, σ^2)

model (0,3) $\epsilon_i = \theta_1\nu_{i-1} + \theta_2\nu_{i-2} + \theta_3\nu_{i-3} + \nu_i$ where ν_i assumed NID(o, σ^2)

```
> library(nlme)
> arma.0.0 <- gls(price ~ landing, data = a, method = "ML")
> arma.1.0 <- gls(price ~ landing, data = a, correlation = corARMA(p = 1,
+     q = 0), method = "ML")
> arma.2.0 <- gls(price ~ landing, data = a, correlation = corARMA(p = 2,
+     q = 0), method = "ML")
> arma.3.0 <- gls(price ~ landing, data = a, correlation = corARMA(p = 3,
+     q = 0), method = "ML")
> arma.0.1 <- gls(price ~ landing, data = a, correlation = corARMA(p = 0,
+     q = 1), method = "ML")
```

```
> arma.1.1 <- gls(price ~ landing, data = a, correlation = corARMA(p = 1,
+     q = 1), method = "ML")
> arma.0.2 <- gls(price ~ landing, data = a, correlation = corARMA(p = 0,
+     q = 2), method = "ML")
> arma.1.2 <- gls(price ~ landing, data = a, correlation = corARMA(p = 1,
+     q = 2), method = "ML")
> arma.0.3 <- gls(price ~ landing, data = a, correlation = corARMA(p = 0,
+     q = 3), method = "ML")
```

The models gives the following statistics where "Lag n" relates to the Durbin-Watson statistic of the residual with lag n

	Par	LogLik	Sigma	Lag 1	Lag 2	Lag 3	Lag 4
Model (0,0)	2	-17.5485	0.5027	0.5832	1.0201	0.7450	0.9012
Model (1,0)	3	-10.3114	0.3497	1.7281	2.8730	1.1315	0.9655
Model (0,1)	3	-7.4173	0.2838	1.1686	1.8289	1.1204	0.9438
Model (2,0)	4	-10.0513	0.3505	1.7435	2.7329	1.0537	0.9333
Model (1,1)	4	-5.1579	0.2617	1.8248	2.4718	1.2500	0.9732
Model (0,2)	4	-3.8654	0.2251	1.8856	1.8345	1.3978	1.3758
Model (1,2)	5	-5.9966	0.3949	0.8986	1.4164	1.0768	1.2005
Model (0,3)	5	-3.7112	0.2167	1.8340	1.7615	1.4962	1.3892

Model (0,0) show autocorrelation for lag 1 and lag 2. In improving this model with one more parameter the model (0,1), in compar with model (1,0), shows the highest likelihood and the smallest σ. However the model (0,1) still have autocorrelation and the model (1,0) have a negative autocorrelation for lag 2. Improvement of model (1,0) with one more autocorrelation term do not seem to yield a good result. When improving model (0,1) with one more parameter, model (0,2) shows a higher likelihood and lower σ than model (1,1), all Durbin-Watson statistics is better for model (0,2) too, hence model (0,2) is preferred for the models with 4 parameters.

There seems to be no gain in adding one more parameter, the best model here is model (0,3), and her the likelihood is only slightly improved. Model (0,2) is accepted as final model.

Parameter estimates for the model (0,2) is:

Parameter	Estimate	Std.error	t-value	p-value
θ_1	1.9908			
θ_2	1.0000			
α	4.0104	0.2517	15.93	1.447e-13
β	-0.01309	0.001223	-10.70	3.473e-10

Both parrameters is very significant.

3 Conclusion

The price in the future (for time $t + 1$) can be predicted by

$$E(p_{t+1}) = \alpha + \beta h_{t+1} + \theta_1 \nu_t + \theta_2 \nu_{t-1}$$

Where the parameters $(\alpha, \beta, \theta_1, \theta_2)$ = (4.0104, -0.01309, 1.9908, 1.0000). The formula for the expected price for next year $E(p_{t+1})$ consist of three parts: The first, $\alpha + \beta h_{t+1}$ is a linear function of next years harvest h_{t+1}, the next $\theta_1 \nu_t$ is a correction probational to this years error term ν_t and the last, $\theta_2 \nu_{t-1}$ is a correction probational to last years error term ν_{t-1}. The ν_i's needed for making prediction is estimated (so that var(ν) is minimize), and are given in the text file `price_herring.txt`.

```
> a$nu <- v0.2
> write.table(a, file = "price_herring.txt")
```

References

Arnason, R., L. K. Sandal, S.I. Steinshamn, N. Vestergaard, S. Argansson and F. Jensen (2000). Comparative evaluation of the cod and herring fisheries in denmark, iceland and norway. I: *TemaNord 2000:526*. Nordisk Ministerråd.

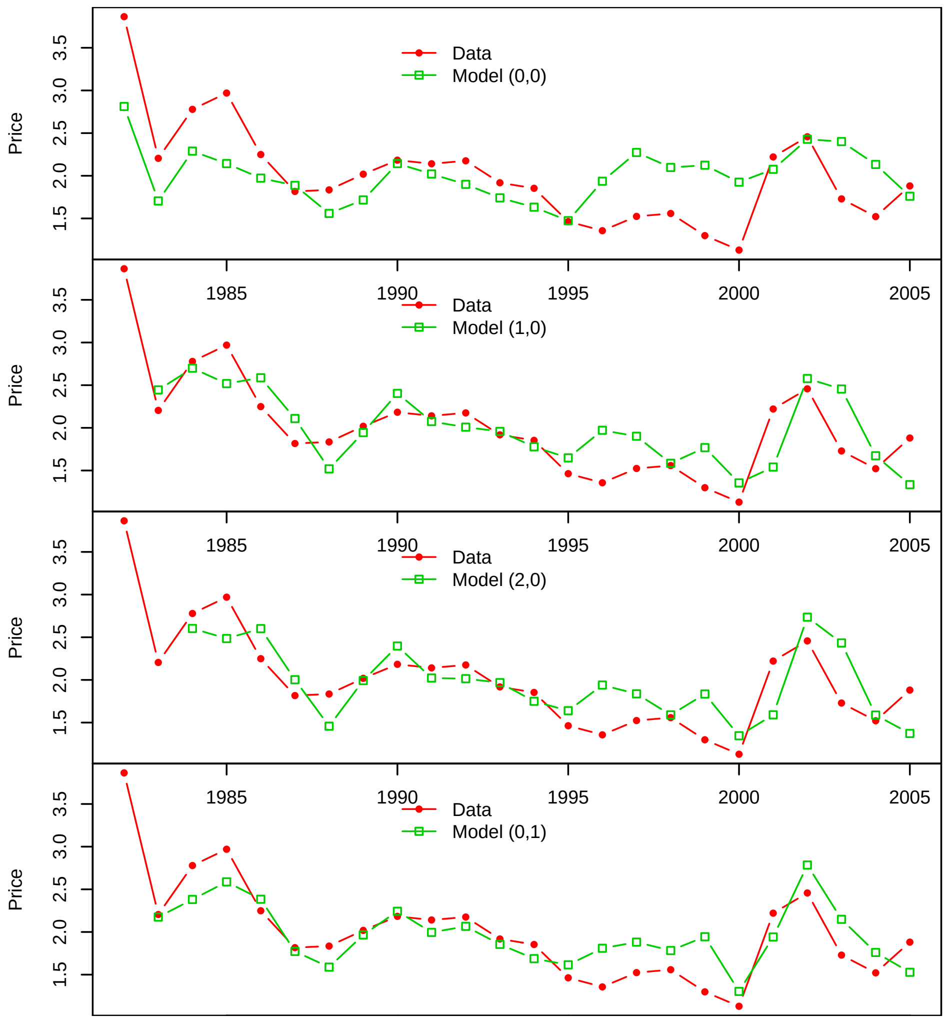
Danmarks Statistik (2006). *PRIS8: Forbrugerprisindeks, årsgennemsnit (1900=100)*. Danmarks Statistik. `http://www.statistikbanken.dk`

Fiskeridirektoratet (2006*a*). *Danske fiskeres fangster fra samtlige farvande fordelt på arter 1996-2005, Hel fisk i ton*. Fiskeridirektoratet. Quoted 20/6 2006. `http://webfd.fd.dk/stat/Faste%20tabeller/Landinger-10aar/tab74b.html`

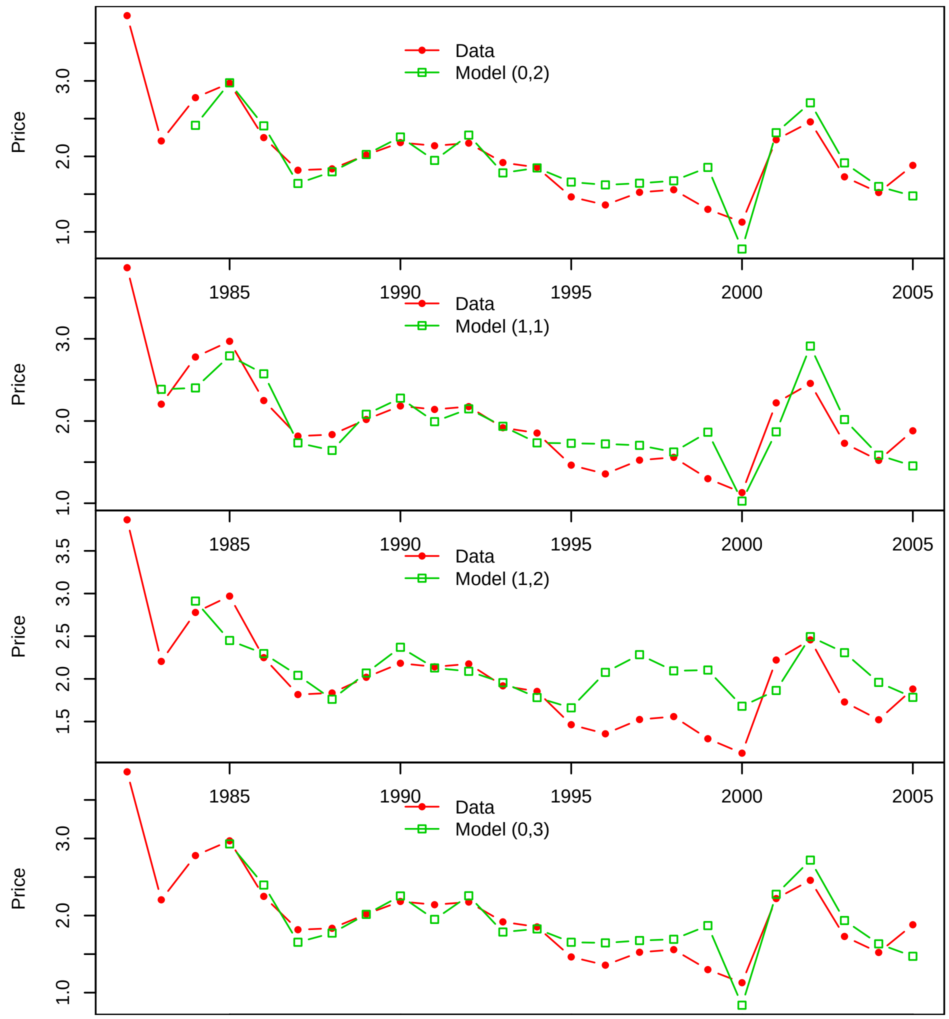
Fiskeridirektoratet (2006*b*). *Danske fiskeres fangster fra samtlige farvande fordelt på arter 1996-2005, Værdi i 1.000 kr.*. Fiskeridirektoratet. Quoted 20/6 2006. `http://webfd.fd.dk/stat/Faste%20tabeller/Landinger-10aar/tab74b.html`

Appendix

Figures of data and model predictions



Figures of data and model predictions



Summary of model (0,2):

```
> summary(arma.0.2)

Generalized least squares fit by maximum likelihood
  Model: price ~ landing
  Data: a
       AIC      BIC    logLik
  17.73089 23.62116 -3.865445

Correlation Structure: ARMA(0,2)
 Formula: ~1
 Parameter estimate(s):
  Theta1   Theta2
1.990781 0.999982

Coefficients:
                Value  Std.Error   t-value p-value
(Intercept)  4.010429 0.25167878  15.93471       0
landing     -0.013088 0.00122313 -10.70047       0

 Correlation:
        (Intr)
landing -0.656

Standardized residuals:
        Min          Q1         Med          Q3         Max
-1.60818839 -1.11326488  0.09143215  0.37655387  1.61419067

Residual standard error: 0.5664258
Degrees of freedom: 24 total; 22 residual
```

Code chunks

Function to get the coefficients of the ARMA covariation

```
> phi <- function(x) {
+     a <- coef(x$modelStruct, unconstrained = FALSE)
+     b <- NULL
+     d <- NULL
+     if (length(a) != 0) {
+         for (i in 1:length(a)) {
+             b <- c(b, a[[i]])
+         }
+         for (i in 1:length(a)) {
+             d <- c(d, paste("Phi", i, sep = ""))
+         }
```

```
+         names(b) <- d
+     }
+     return(b)
+ }
```

Prediction and residuals for model (0,0)

```
> rp <- ts(start = 1982, end = 2005, a$price)
> h <- ts(start = 1982, end = 2005, a$landing)
> f0 <- ts(start = 1982, end = 2005, fitted(arma.0.0))
> t0 <- ts.union(rp = rp, h = h, pre = f0, risd = rp - f0)
```

Prediction and residuals for model (1,0)

```
> r1 <- ts(start = 1982, end = 2005, a$price - fitted(arma.1.0))
> f1 <- ts(start = 1982, end = 2005, fitted(arma.1.0))
> phi1 <- phi(arma.1.0)
> t1 <- ts.union(rp = rp, h = h, pre = f1 + phi1 * lag(r1, -1),
+     risd = rp - (f1 + phi1 * lag(r1, -1)))
```

Prediction and residuals for model (2,0)

```
> r2 <- ts(start = 1982, end = 2005, a$price - fitted(arma.2.0))
> f2 <- ts(start = 1982, end = 2005, fitted(arma.2.0))
> phi1 <- phi(arma.2.0)[1]
> phi2 <- phi(arma.2.0)[2]
> t2 <- ts.union(rp = rp, h = h, pre = f2 + phi1 * lag(r2, -1) +
+     phi2 * lag(r2, -2), risd = rp - (f2 + phi1 * lag(r2, -1) +
+     phi2 * lag(r2, -2)))
```

Prediction and residuals for model (0,1)

```
> r0.1 <- ts(start = 1982, end = 2005, a$price - fitted(arma.0.1))
> f0.1 <- ts(start = 1982, end = 2005, fitted(arma.0.1))
> theta <- phi(arma.0.1)
> v0.1 <- ts(start = 1982, end = 2005, NA)
> St <- function(a) {
+     v0.1[1] <- a
+     for (i in 2:length(v0.1)) {
+         v0.1[i] <- r0.1[i] - theta * v0.1[i - 1]
+     }
+     return(var(v0.1))
+ }
> v0.1[1] <- optimize(f = St, interval = c(-1, 1))[[1]]
> for (i in 2:length(v0.1)) {
+     v0.1[i] <- r0.1[i] - theta * v0.1[i - 1]
+ }
> t0.1 <- ts.union(rp = rp, h = h, pre = f0.1 + theta * lag(v0.1,
+     -1), risd = rp - (f0.1 + theta * lag(v0.1, -1)))
```

Prediction and residuals for model (1,1)

```
> r1.1 <- ts(start = 1982, end = 2005, a$price - fitted(arma.1.1))
> f1.1 <- ts(start = 1982, end = 2005, fitted(arma.1.1))
> phi1 <- phi(arma.1.1)[1]
> theta <- phi(arma.1.1)[2]
> v1.1 <- ts(start = 1982, end = 2005, NA)
> St <- function(a) {
+     v1.1[1] <- a
+     for (i in 2:length(v1.1)) {
+         v1.1[i] <- r1.1[i] - phi1 * r1.1[i - 1] - theta * v1.1[i -
+             1]
+     }
+     return(var(v1.1))
+ }
> v1.1[1] <- optimize(f = St, interval = c(-1, 1))[[1]]
> for (i in 2:length(v1.1)) {
+     v1.1[i] <- r1.1[i] - phi1 * r1.1[i - 1] - theta * v1.1[i -
+         1]
+ }
> t1.1 <- ts.union(rp = rp, h = h, pre = f1.1 + theta * lag(v1.1,
+     -1) + phi1 * lag(r1.1, -1), risd = rp - (f1.1 + theta * lag(v1.1,
+     -1) + phi1 * lag(r1.1, -1)))
```

Prediction and residuals for model (0,2)

```
> r0.2 <- ts(start = 1982, end = 2005, a$price - fitted(arma.0.2))
> f0.2 <- ts(start = 1982, end = 2005, fitted(arma.0.2))
> theta1 <- phi(arma.0.2)[1]
> theta2 <- phi(arma.0.2)[2]
> v0.2 <- ts(start = 1982, end = 2005, NA)
> St <- function(a) {
+     v0.2[1] <- a[1]
+     v0.2[2] <- a[2]
+     for (i in 3:length(v0.2)) {
+         v0.2[i] <- r0.2[i] - theta1 * v0.2[i - 1] - theta2 *
+             v0.2[i - 2]
+     }
+     return(var(v0.2))
+ }
> v0.2[1:2] <- optim(c(1, 1), St)$par
> for (i in 3:length(v0.2)) {
+     v0.2[i] <- r0.2[i] - theta1 * v0.2[i - 1] - theta2 * v0.2[i -
+         2]
+ }
> t0.2 <- ts.union(rp = rp, h = h, pre = f0.2 + theta1 * lag(v0.2,
```

```
+     -1) + theta2 * lag(v0.2, -2), risd = rp - (f0.2 + theta1 *
+     lag(v0.2, -1) + theta2 * lag(v0.2, -2)))
```

Prediction and residuals for model (1,2)

```
> r1.2 <- ts(start = 1982, end = 2005, a$price - fitted(arma.1.2))
> f1.2 <- ts(start = 1982, end = 2005, fitted(arma.1.2))
> phi1 <- phi(arma.1.2)[1]
> theta1 <- phi(arma.1.2)[2]
> theta2 <- phi(arma.1.2)[3]
> v1.2 <- ts(start = 1982, end = 2005, NA)
> St <- function(a) {
+     v1.2[1] <- a[1]
+     v1.2[2] <- a[2]
+     for (i in 3:length(v1.2)) {
+         v1.2[i] <- r1.2[i] - phi1 * r1.2[i - 1] - theta1 * v1.2[i -
+             1] - theta2 * v1.2[i - 2]
+     }
+     return(var(v1.2))
+ }
> v1.2[1:2] <- optim(c(1, 1), St)$par
> for (i in 3:length(v1.2)) {
+     v1.2[i] <- r1.2[i] - theta1 * v1.2[i - 1] - theta2 * v1.2[i -
+         2]
+ }
> t1.2 <- ts.union(rp = rp, h = h, pre = f1.2 + phi1 * lag(r1.2,
+     -1) + theta1 * lag(v1.2, -1) + theta2 * lag(v1.2, -2), risd = rp -
+     (f1.2 + phi1 * lag(r1.2, -1) + theta1 * lag(v1.2, -1) + theta2 *
+         lag(v1.2, -2)))
```

Prediction and residuals for model (0,3)

```
> r0.3 <- ts(start = 1982, end = 2005, a$price - fitted(arma.0.3))
> f0.3 <- ts(start = 1982, end = 2005, fitted(arma.0.3))
> theta1 <- phi(arma.0.3)[1]
> theta2 <- phi(arma.0.3)[2]
> theta3 <- phi(arma.0.3)[3]
> v0.3 <- ts(start = 1982, end = 2005, NA)
> St <- function(a) {
+     v0.3[1] <- a[1]
+     v0.3[2] <- a[2]
+     v0.3[3] <- a[3]
+     for (i in 4:length(v0.3)) {
+         v0.3[i] <- r0.3[i] - theta1 * v0.3[i - 1] - theta2 *
+             v0.3[i - 2] - theta3 * v0.3[i - 3]
+     }
+     return(var(v0.3))
```

```
+ }
> v0.3[1:3] <- optim(c(1, 1, 1), St)$par
> for (i in 4:length(v0.3)) {
+     v0.3[i] <- r0.3[i] - theta1 * v0.3[i - 1] - theta2 * v0.3[i -
+         2] - theta3 * v0.3[i - 3]
+ }
> t0.3 <- ts.union(rp = rp, h = h, pre = f0.3 + theta1 * lag(v0.3,
+     -1) + theta2 * lag(v0.3, -2) + theta3 * lag(v0.3, -3), risd = rp -
+     (f0.3 + theta1 * lag(v0.3, -1) + theta2 * lag(v0.3, -2) +
+         theta3 * lag(v0.3, -3)))
```

Function to calculate Durbin-Watson statistic

```
> dw <- function(e) {
+     a <- 0
+     for (t in 2:length(e)) {
+         a <- a + (e[t] - e[t - 1])^2
+     }
+     return(a/sum(e^2))
+ }
> dw2 <- function(e) {
+     a <- 0
+     for (t in 3:length(e)) {
+         a <- a + (e[t] - e[t - 2])^2
+     }
+     return(a/sum(e^2))
+ }
> dw3 <- function(e) {
+     a <- 0
+     for (t in 4:length(e)) {
+         a <- a + (e[t] - e[t - 3])^2
+     }
+     return(a/sum(e^2))
+ }
> dw4 <- function(e) {
+     a <- 0
+     for (t in 5:length(e)) {
+         a <- a + (e[t] - e[t - 4])^2
+     }
+     return(a/sum(e^2))
+ }
```

Functions to extract the information for the table

```
> pn <- function(x, d = 4) {
+     round(x, d)
+ }
```

```
> pri <- function(a) {
+     c(Par = pn(length(c(summary(a)$coefficients, phi(a)))), LogLik = pn(a$logLik))
+ }
> rse <- function(x) {
+     return(sqrt(sum(x^2)/length(x)))
+ }
> lagg <- function(a) {
+     c(Sigma = pn(rse(a)), "Lag 1" = pn(dw(a)), "Lag 2" = pn(dw2(a)),
+         "Lag 3" = pn(dw3(a)), "Lag 4" = pn(dw4(a)))
+ }
```

Code for the table

```
> rbind("Model (0,0)" = c(pri(arma.0.0), lagg(t0[, 4])), "Model (1,0)" = c(pri(arma.1.0),
+     lagg(t1[-1, 4])), "Model (0,1)" = c(pri(arma.0.1), lagg(t0.1[-1,
+     4])), "Model (2,0)" = c(pri(arma.2.0), lagg(t2[-(1:2), 4])),
+     "Model (1,1)" = c(pri(arma.1.1), lagg(t1.1[-1, 4])), "Model (0,2)" = c(pri(arma.0.2),
+         lagg(t0.2[-(1:2), 4])), "Model (1,2)" = c(pri(arma.1.2),
+         lagg(t1.2[-(1:2), 4])), "Model (0,3)" = c(pri(arma.0.3),
+         lagg(t0.3[-(1:3), 4])))
```

Code for figure 1

```
> par(mfcol = c(4, 1), mar = c(0, 5.1, 0, 2.1))
> plot(t0[, c(1, 3)], plot.type = "single", type = "b", col = c(2,
+     3), pch = c(20, 22), ylab = "Price")
> legend(1990, 3.7, c("Data", "Model (0,0)"), lty = 1, col = 2:3,
+     pch = c(20, 22), merge = T, bty = "n")
> plot(t1[, c(1, 3)], plot.type = "single", type = "b", col = c(2,
+     3), pch = c(20, 22), ylab = "Price")
> legend(1990, 3.7, c("Data", "Model (1,0)"), lty = 1, col = 2:3,
+     pch = c(20, 22), merge = T, bty = "n")
> plot(t2[, c(1, 3)], plot.type = "single", type = "b", col = c(2,
+     3), pch = c(20, 22), ylab = "Price")
> legend(1990, 3.7, c("Data", "Model (2,0)"), lty = 1, col = 2:3,
+     pch = c(20, 22), merge = T, bty = "n")
> plot(t0.1[, c(1, 3)], plot.type = "single", type = "b", col = c(2,
+     3), pch = c(20, 22), ylab = "Price")
> legend(1990, 3.7, c("Data", "Model (0,1)"), lty = 1, col = 2:3,
+     pch = c(20, 22), merge = T, bty = "n")
```

Code for figure 2

```
> par(mfcol = c(4, 1), mar = c(0, 5.1, 0, 2.1))
> plot(t0.2[, c(1, 3)], plot.type = "single", type = "b", col = c(2,
+     3), pch = c(20, 22), ylab = "Price")
> legend(1990, 3.7, c("Data", "Model (0,2)"), lty = 1, col = 2:3,
```

```
+     pch = c(20, 22), merge = T, bty = "n")
> plot(t1.1[, c(1, 3)], plot.type = "single", type = "b", col = c(2,
+     3), pch = c(20, 22), ylab = "Price")
> legend(1990, 3.7, c("Data", "Model (1,1)"), lty = 1, col = 2:3,
+     pch = c(20, 22), merge = T, bty = "n")
> plot(t1.2[, c(1, 3)], plot.type = "single", type = "b", col = c(2,
+     3), pch = c(20, 22), ylab = "Price")
> legend(1990, 3.7, c("Data", "Model (1,2)"), lty = 1, col = 2:3,
+     pch = c(20, 22), merge = T, bty = "n")
> plot(t0.3[, c(1, 3)], plot.type = "single", type = "b", col = c(2,
+     3), pch = c(20, 22), ylab = "Price")
> legend(1990, 3.7, c("Data", "Model (0,3)"), lty = 1, col = 2:3,
+     pch = c(20, 22), merge = T, bty = "n")
```