STAT 804 Solutions

Assignment 3

1. The Splus version 3 command

   attach('/home/math4/lockhart/Teaching/courses/804/datasets')

   or the Splus version 5 command

   attach('/home/math4/lockhart/Teaching/courses/804/datasetsS5')

   will make a dataset *influenza* available. If you type

   ls(pos=2)

   you will see the data set for this question and the next two.

   The data consist of monthly counts of influenza cases over a 9 and a half year period. Fit an ARIMA model to the data.

   Solution: My analysis is distributed separately.

2. Fit an ARIMA model for the Johnson and Johnson earnings per share data which is in the dataset *earnings*. There are 20 years of quarterly data. **Solution:** My personal favourite model is obtained by taking logs first and then, if X_t is the series of logarithms, fitting the model

   $$Y_t = \alpha_0 + \alpha_1 t + W_t$$

   where W is an ARIMA(1,0,0)$\times(1,0,0)_4$ series whose fitted model is

   $$(I - 0.8B^4)(I - 0.266B)W = \epsilon$$

   with ϵ a white noise series with variance 0.0065. The fitted values of α_0 and α_1 are -0.675 and 0.0415; SPlus does not provide standard errors. The standard errors in the AR coefficients are 0.069 (for the 0.8 value) and 0.11 for the 0.266 value so that both are significant.

   Many students fitted a model with one or more differences taken at lag 4. I found it quite hard to get a really good fit this way. There is an important conceptual difference between fitting the linear trend as I have and taking differences. The differencing method supposes that earnings respond more or less directly to earlier values of earnings whereas I suppose the linear trend d to arise as a result of an external driving force pushing the earnings up exponentially. Moreover the differenced models require increasing variance with time; my model is stationary about the linear trend.

   I accepted a wide variety of answers, grading on the basis of how convincing and well reasoned I thought your process was. Generally I would prefer to see t hat fitted models were presented more or less as above, complete with clear formulas parameter estimates and standard errors.

3. Fit an ARIMA model for the data set called *fake*.

4. For the data set *raindiff* fit the model

$$X_t - \mu = \phi(X_{t-1} - \mu) + \epsilon_t$$

where the ϵ_T are iid $N(0, \sigma^2)$ in each of the following ways

(a) By full maximum likelihood. (Maximize the joint density of the X's over the parameters μ, ϕ and σ.)

(b) By estimating μ by the sample mean and then doing full maximum likelihood for the model

$$Y_t = \phi Y_{y-1} + \epsilon_t$$

where $Y_t = X_t - \bar{X}$.

(c) By estimating μ by the sample mean and then doing conditional maximum likelihood for the previous model.

Remarks: I do not want you to use *ar* or other built-in S time-series functions to do these but you may want to use them to get starting values for estimates. It is probably easier to do the methods in reverse order, using the results as starting points for iterative solutions.

Solution: The conditional log-likelihood in the 3rd part of this problem is

$$\frac{-1}{2\sigma^2}\sum(y_t - \phi y_{t-1})^2$$

The resulting score has components

$$\frac{-1}{\sigma^2}(\sum_1^{T-1} y_t y_{t-1} - \phi \sum_0^{T-2} y_t^2$$

and

$$\frac{-1}{\sigma^3}\sum(y_t - \phi y_{t-1})^2 - (T-1)/\sigma\,.$$

The estimate of ϕ is $\sum_1^{T-1} y_t y_{t-1} / \sum_0^{T-2} y_t^2$ and that of σ^2 is $\sum(y_t - \hat{\phi} y_{t-1})^2/(T-1)$

Thus the answers to part 3 are $\hat{\phi} = 0.5572891$, $\hat{\sigma} = 3.616531$ and $\hat{\mu} = -0.588764$. (All are given to many decimal places only for comparison purposes.)

In part 2 the log-likelihood is augmented by another $-\log(\sigma)$, by $\log(1-\phi^2)/2$ and by the added term $-(1-\phi^2)y_0^2/(2\sigma^2)$. The two components of the score are now

$$\frac{\sum(y_t - \phi y_{t-1})^2 + (1-\phi^2)y_0^2}{\sigma^3} - T/\sigma$$

and

$$\frac{\sum(y_t y_{t-1} - \phi y_{t-1}^2) - \phi y_0^2}{\sigma^2} - \frac{\phi}{1-\phi^2}\,.$$

Now

$$\hat{\sigma^2} = \frac{\sum(y_t - \hat{\phi}y_{t-1})^2 + (1 - T\phi^2)y_0^2}{T}.$$

To find $\hat{\phi}$ we plug this formula for $\hat{\sigma}$ into the log likelihood and obtain the *profile* likelihood for ϕ,

$$-\frac{T}{2} - T\log(\hat{\sigma}) + \log(1 - \phi^2)/2$$

where now you must keep in mind that $\hat{\sigma}$ depends on ϕ. I used the following S-Plus functions:

```
tau0 <-function(x, phi)
{
        n <- length(x)
        m <- outer(rep(1, length(phi)), x[-1]) - outer(phi, x[ - n])
        t1 <- m^2 %*% rep(1, n - 1)
        (t1 + (1 - phi^2) * x[1]^2)/n
}
ell0 <- function(phi)
{
        n <- length(x0)
        n * log(tau0(x0, phi)) - log(1 - phi^2)/2
}
nlmin(ell0,0.55)
```

I had created the data set `x0 <- raindiff - mean(raindiff)` ahead of time because this is a simple way to pass the data to the function `ell0` which is to be minimized by `nlmin`.

For part 2 then we have $\hat{\phi} = 0.5592308$, $\hat{\sigma} = 3.6125532$ and $\hat{\mu} = -0.588764$.

For part 1 the y's of part 2 are replaced by $x - \mu$ and the μ component of the score function can be set equal to 0 to find

$$\hat{\mu} = \frac{\sum_0^{T-1} X_t - \phi\sum_0^{T-2} X_t + \phi X_0}{n - (n-2)\phi}$$

I used

```
tau <-function(x, phi)
{
        y <- t(outer(x, mu(x, phi), "-"))
        n <- length(x)
        m <- y[, -1] - phi * y[,  - n]
        t1 <- m^2 %*% rep(1, n - 1)
        (t1 + (1 - phi^2) * y[, 1]^2)/n
}
```

```
ell <- function(phi)
{
        n <- length(y)
        n * log(tau(y, phi)) - log(1 - phi^2)/2
}
nlmin(ell,0.55)
```

Again the data set is `y <- raindiff`. Notice the use of the outer function to fill a matrix with $x_j - \phi_i$.

For part 3 I get $\hat{\phi} = 0.5592308$, $\hat{\sigma} = 3.6125531$ and $\hat{\mu} = -0.5904616$.

Note on SPlus: I used the following function to call nlmin and to make the data set available to the functions ell and ell0:

```
estim.ar1 <- function(x)
{
   n <- length(x)
   m <- mean(x)
   assign("y", x, 1)
   assign("y0", x - m, 1)
   phi.hat.3 <- sum(y0[-1] * y0[ - n])/sum(y0[ - n]^2)
   sigma.hat.3 <- sqrt(sum((y0[-1] - phi.hat.3 * y0[ - n])^2)/(n - 1))
   phi.hat.2 <- nlmin(ell0, phi.hat.3)$x
   sigma.hat.2 <- sqrt(tau0(y0, phi.hat.2))
   phi.hat.1 <- nlmin(ell, phi.hat.2)$x
   sigma.hat.1 <- sqrt(tau(y, phi.hat.1))
   mu.hat.1 <- mu(y, phi.hat.1)
   matrix(c(m, phi.hat.3, sigma.hat.3, m, phi.hat.2, sigma.hat.2,
      mu.hat.1, phi.hat.1, sigma.hat.1), nrow = 3)
}
```

The function `assign` makes the variables y and $y0$ available to functions called by `estim.ar1` without creating them permanently. This permits `ell` and `ell0` to access y and $y0$ even though nlmin does not permit any argument but ϕ to be passed through the argument list of `ell` or `ell0`.

Overall there is very little point in the more difficult computations; these estimates agree to a small fraction of a standard error.