Introduction to R for Times Series Analysis

G.P. Nason

28th October 1994
Revised 7th November 1997, 28th September 2005.

1 Using R

About R. R is a increasingly popular statistical package. It has very good graphics facilities and is very flexible. The work I will set you during the time series course will require no previous knowledge of R, however, you will have already encountered it at sometime during your time here. R is a freeware package and so you could download it to your personal computer (versions for Linux and the Mac exist, as well as for Windows). More help on R and downloads can be found at the comprehensive R archive network (CRAN)

```
http://cran.r-project.org
```

Help on R. There is a large amount of information available about R. The CRAN website has a lot of help. Additionally, R has its own help system. Once in R all you have to type is a question mark followed by the command you want help on, e.g. if you want the help on he `ls` command just type

```
> ?ls
```

and a help window will appear.

Using R in the lab. The easiest way is to locate the R icon on your desktop and double-click it. After starting you will see a large window containing a smaller window called the commands window where you type all your commands to R.

If you want to exit R at any time either type

```
> q()
```

in the commands window or select `Exit` off the `File` menu.

You can repeat and edit previous commands by using the up- and down-arrow keys.

We have arranged for R to store its working files in the subdirectory R in your `cufs` filespace (the files are hidden but you can see them by changing the view options if you really want to). In particular, if you `save()`, `dump()` or `write()` any information from R to the filesystem it will appear in the R subdirectory. You can look at the R help on these functions if you want).

1.1 Vector operations

R stores simple data in vector and matrices and performs most operations directly on these. Therefore looping operations that process individual array elements are rare. For

example, the following FORTRAN[1] code computes the mean of 10 data points in the array X

```
    XBAR = 0.0
    DO 10 I=1,N
        XBAR = XBAR + X(I)
10  CONTINUE
    XBAR = XBAR/N
```

can be written in R using the succinct notation

```
> xbar <- sum(x)/n
```

Many common statistical operations are already coded into R functions. The `mean()` function is one of these and so the previous R code can be replaced by

```
> xbar <- mean(x)
```

You've probably noticed by now that the assignment is made using `<-` (not `=`, although it can be used) and that the `>` symbol is S-Plus's prompt.

2 Time-series analyses

We are going to use R is to perform time-series analyses. We will give several structured examples.

2.1 Fitting an AR model

The `lynx` data set is already available to you. The data set refers to the number of Canadian lynx trapped each year from 1821 until 1934. Type

```
> lynx
```

to see the data. Notice how R puts the reference dates down the left hand-side. To plot this data type

```
> ts.plot(lynx)
```

You will see a strongly periodic pattern with sharp peaks every 10 years or so. To confirm this periodic period have a look at the autocorrelation function of the lynx data. This is performed by the

```
> acf(lynx)
```

command. What do you think? The time-series plot indicates that the data is not stationary. Try taking logs of the lynx data and then plotting it.

```
> ts.plot(log(lynx))
```

and then do the same for the acf

```
> acf(log(lynx))
```

[1] Even if you don't know FORTRAN you should be able to work out what this code segment does

Use the help facility to find the help page for the `acf()` function (just type `?acf`) and try changing some of the arguments to the function. In particular, try

```
> acf(log(lynx), type="partial")
```

What picture do you get? Is there a cut-off value past which the partial autocorrelation coefficients are zero? What do the dotted lines on the plot mean (look at the help page)?

Now fit an AR model using `ar.yw()`

```
> llynx.ar <- ar.yw(log(lynx))
```

The function `ar.yw()` returns a *composite object* which we put into `llynx.ar`. The object contains many pieces of information about the fitted AR model. You can get the names of all the components in the `llynx.ar` object by typing

```
> names(llynx.ar)
```

In particular, the `order.max` component specifies the maximum order model that is considered in the fit. To look at the value of this type

```
> llynx.ar$order.max
```

and you will see that in this case it was 20 (this can be changed as an argument to `ar.yw()`). The `$` character is used to access parts of a composite object.

By default the `ar.yw()` function uses Akaike's information criterion to decide which model to fit. This information is stored in the `llynx.ar` object and you can view the AIC for all values of p, the order of all the AR models considered, by typing

```
> ts.plot(llynx.ar$aic, main="AIC for Log(Lynx)")
```

From the plot you will see that the lowest part is at $p = 11$. This is also the value stored in the `order` component of `llynx.ar` (check this by typing

```
> llynx.ar$order
```

and seeing that it is indeed 11.

Type

```
> llynx.ar$ar
```

and you will see a list of the coefficients that we fitted for the order 11 model. Therefore the model fitted was

$$X_t = 1.14X_{t-1} - 0.51X_{t-2} + \ldots - 0.31X_{t-11} + Z_t$$

where I have rounded the coefficients to 2 d.p. R makes it very easy to see the fit. Type

```
> ts.plot(log(lynx) - llynx.ar$resid)
> lines(log(lynx), col=2)
```

The black line is the fit (achieved by removing the residuals from the original data because

$$\text{residual} = \text{data} - \text{fit})$$

and the coloured line is the original data (the `col=2` argument causes the line to be drawn in blue).

3 Fitting an ARIMA model in R

The data described in this section are held in a matrix but in my web space. R makes it very easy to access data in other places. To enable access for this data type the following (all on one line, there should not be a new line between `magpn/` and `Teaching`, it is there so I can fit it on the page)

```
> load(url("http://www.stats.bris.ac.uk/~magpn/
   Teaching/TimeSeries/Data/wool.RData"))
```

Feel free to use the help facility to obtain information about `load(), url()` and related functions.

The data you have just loaded is held in a matrix called `wool`. This is a matrix containing 310 separate observations on 10 variables. The `dim()` function tells you this:

```
> dim(wool)
[1] 310  10
```

The variables are described in Table 1. You can see the variable names in R by typing

Variable No.	Description
1	Index number
2	Calendar year (1976-84)
3	Calendar week (1-52)
4	Weeks since 1.1.76
5	Floor price (cents per kg)
6	Actual price (cents per kg)
7	Ratio of Actual to Floor price
8	Log(Floor price)
9	Log(Actual price)
10	Log(Actual price/Floor price)

Table 1: Variables in the wool data set

```
> dimnames(wool)
```

The wool data set contains prices monitored by the Australian Wool Corporation from June 1976 to June 1984. The prices are monitored weekly with some breaks for public holidays, for example over the Christmas period there is a break of several weeks. Before the start of each week the Corporation sets a floor price for the week. The Corporation guarantees that it will pay this price for the wool during the week. The actual price of the wool is an average taken over the following week and is never less than the floor price (otherwise they could have sold it to the Corporation and made more money). You can see this for yourself from a plot of the 5th and 6th variables listed in Table 1. A plot of these two variables appears in Figure 1. The figure was created using the R command

```
> ts.plot(wool[,5:6], lty=1:2)
```

Note the method of accessing data in the matrix. To access the $(i, j)th$ element of a matrix use `[i,j]`, to obtain the jth column use `[,j]`. We will denote the actual

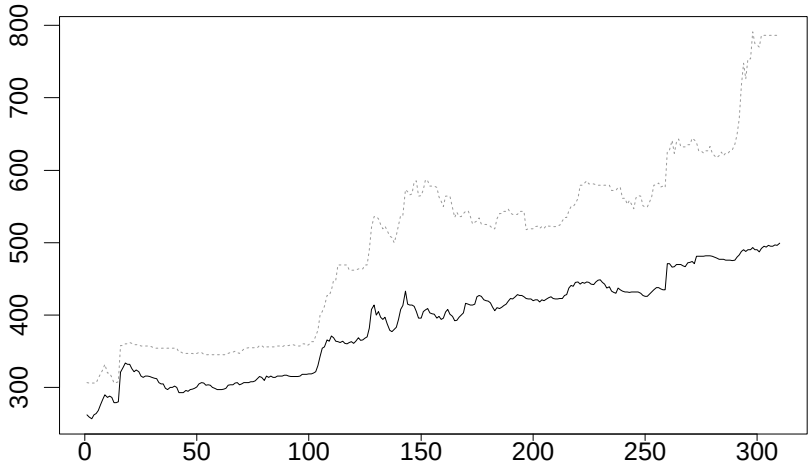

Figure 1: Weekly wool prices. The upper series represents the average weekly price set by the market, the lower series is a floor price set by the Australian Wool Corporation

price by A_t and the floor price by F_t in both cases $t = 1, \ldots, 310$. The interest in this series centres around the Corporation's intervention policy and so it is the relative price movements that we are interested in. Therefore we can look at the ratio of actual to floor price as this will compensate for trends such as those caused by currency fluctuations and inflation. The ratio of actual price to floor price is plotted in Figure 2, we denote the ratio by

$$R_t = \frac{A_t}{F_t} \quad t = 1, \ldots, 310.$$

Another point to consider is that price movements are often multiplicative in nature (so price increases/decreases tend to be discussed in percentage terms rather than absolute terms). We can then feel justified in working with the log of the series which we denote

$$L_t = \log(R_t) \quad t = 1, \ldots, 310$$

rather than the series itself. This is illustrated in Figure 3. The first thing to notice about Figure 3 is that it does not look very different to Figure 2 — it is.

3.1 Fitting an ARIMA model

In this section we investigate the possibility of fitting an ARIMA model to the series. We will ignore all the "missing days" in the series and assume that the data are recorded daily with no gaps. To do this we simply follow our ARIMA fitting model order flowchart (from the lecture notes). The first step is to see whether $\{L_t\}$ looks stationary. Figure 3 shows that $\{L_t\}$ is clearly not stationary, the mean appears to change over time (more precisely you might guess that the mean is piecewise constant, for example you might guess that

$$\mu(t) = \begin{cases} \mu_1 = 0.15 & \text{for } 1 \leq t \leq 110 \\ \mu_2 = 0.25 & \text{for } 111 \leq t \leq 285 \\ \mu_3 = 0.46 & \text{for the rest of the series} \end{cases}$$

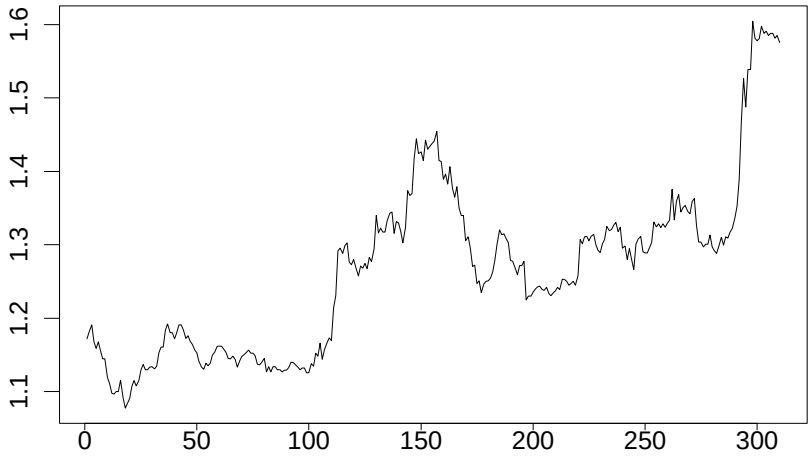

Figure 2: Weekly wool prices. The series represents the ratio of average weekly price set by the market to a floor price set by the Australian Wool Corporation

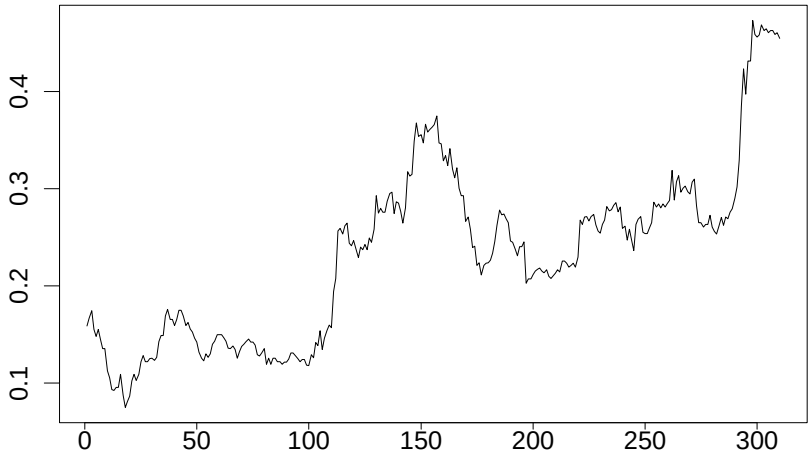

Figure 3: The series represents the log-ratio of average weekly price set by the market to a floor price set by the Australian Wool Corporation

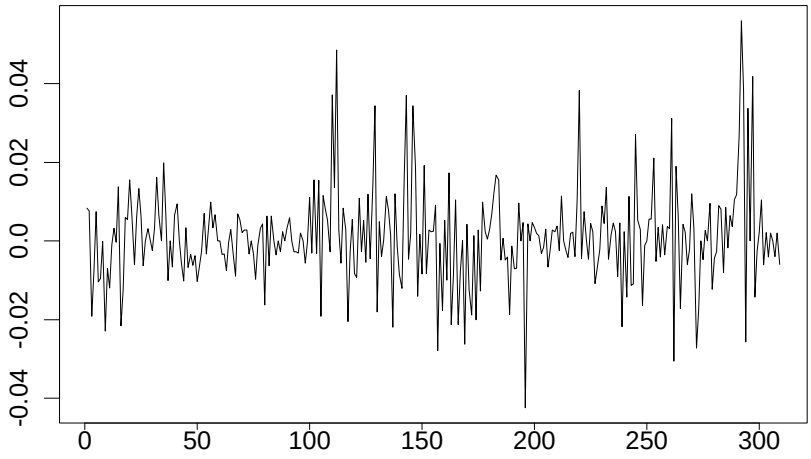

Figure 4: First differences of L_t

approximately). Following the flowchart we difference the series. This is achieved in R by the command

```
> ts.plot(diff(wool[,10]))
```

and produces the series in Figure 4. The striking feature in Figure 4 is that the appears to have been a systematic change in variance of the data after about 100 observations. If we were seriously interested in modelling the series we would have to model the first 100 observations differently to the rest. For this example, we are interested in forecasting future behaviour so we will simply discard the first 100 and model the remaining 209. To simplify matters we will create a new vector called `woolly` that contains the last 209 differenced observations

```
> tmp <- diff(wool[,10])
> woolly <- tmp[101:309]
```

The `:` operator constructs a vector so that `a:b` returns the vector $(a, a+1, a+2, \ldots, b-1, b)$. Note also that although `wool` is a matrix both `tmp` and `woolly` are vectors because the `[,10]` construct extracts a column from the matrix.

The next stage of the flowchart procedure was to produce a correlogram of the differenced data. This is shown in Figure 5. From this it can be seen that only the second and the fourth autocorrelations might possibly be significant. Later we might find it useful to take account of these autocorrelations and fit more complicated models, but in the interests of parsimony we want to fit the simplest model possible. Also, there is no other pattern in the autocorrelations so we conclude that the differenced data is consistent with white noise.

3.2 First model for the wool data

Therefore our first model is

$$L_t - L_{t-1} = \mu + Z_t$$

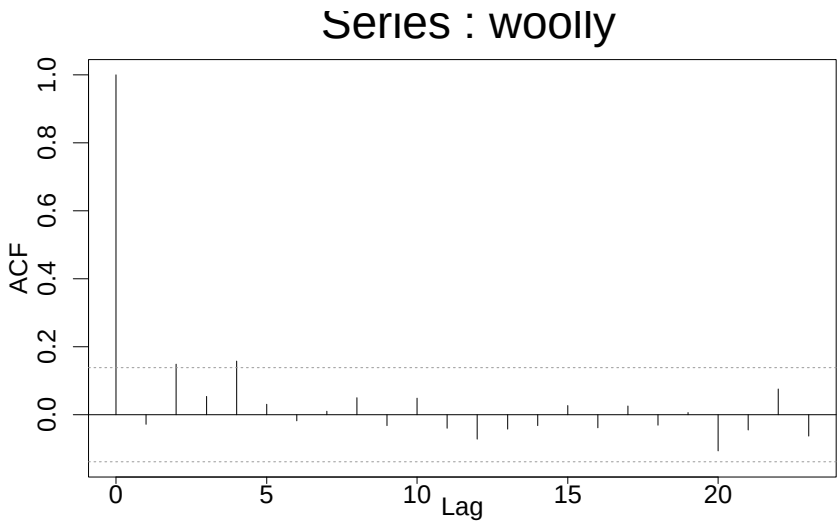

Figure 5: Correlogram of the last 209 first differences of L_t. The dashed horizontal lines correspond to the significance limit $\pm 2/\sqrt{n}$

.

We have a assumed a non-zero mean for this model. Can we do this? Well, the R command

```
> mean(woolly)
[1] 0.001555981
```

suggests that the differences have a non-zero mean. Since we have assumed that the differences are uncorrelated we are justified in using a one-sample t-test to test the hypothesis

$$H_0 : \mu = 0$$

against the alternative

$$H_{\mathrm{A}} : \mu \neq 0$$

This too can be carried out with R by

```
> t.test(woolly)
        One-sample t-Test
data:  woolly
t = 1.6318, df = 208, p-value = 0.1042
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -0.0003238799  0.0034358416
sample estimates:
   mean of x
 0.001555981
```

The p-value suggests that we cannot reject the null hypothesis at the 10% level. So we will assume $\mu = 0$. The R command `var()` allows us to find an estimate of σ_Z^2 the variance of $\{Z_t\}$

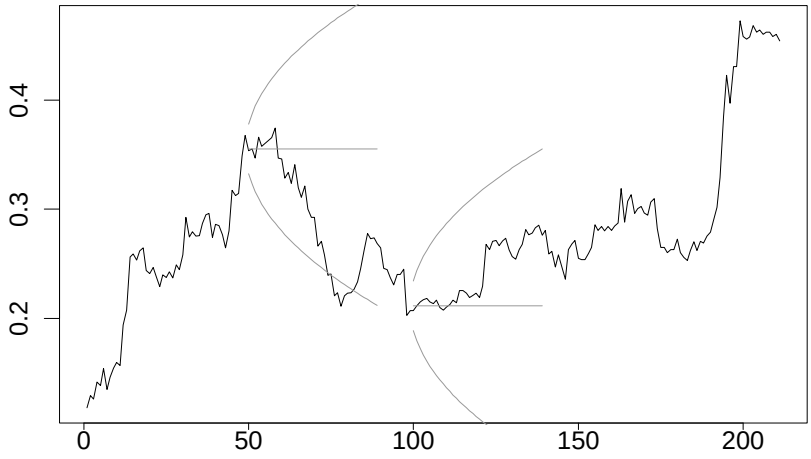

Figure 6: Forecasts for the log-wool-price-ratio. The two horizontal lines are the forecasts for up to 40-steps ahead. The two bowl shapes represent 90% tolerance intervals for the r-step ahead forecasts from $t = 150$ and $t = 200$. The first 100 observations have been discarded.

```
> var(woolly)
[1] 0.0001900347
```

Therefore our second model is

$$L_t = L_{t-1} + Z_t$$

where $\{Z_t\}$ is a purely random process with mean zero and variance $\hat{\sigma}_Z^2 = 0.00019$.

3.3 Forecast errors

The lecture notes give a tolerance interval for forecast errors. The r-step ahead forecast

$$\hat{l}(t, r) = l_t$$

is just the last observation at time t. The tolerance level computed in the lecture notes is

$$L_t \pm 0.0227\sqrt{r}$$

Two of these are plotted in Figure 6 The figure was produced with the R commands

```
> ts.plot(wool[100:310,10])
> lines(50:89, rep(1,40)*wool[150, 10], col=2)
> lines(50:89, wool[150, 10] + 0.0227*sqrt(1:40), col=2)
> lines(50:89, wool[150, 10] - 0.0227*sqrt(1:40), col=2)
> lines(100:139, rep(1,40)*wool[200, 10] , col=2)
> lines(100:139, wool[200, 10] + 0.0227*sqrt(1:40), col=2)
> lines(100:139, wool[200, 10] - 0.0227*sqrt(1:40), col=2)
```

Note that after about 30-steps ahead the first tolerance limit is broken by the series going through a steeply declining phase, although the series remains in the bounds of the second tolerance interval. This is to be expected for such a large number of steps ahead.