Time Series III

Computing Practical 2

Week 3, 2004

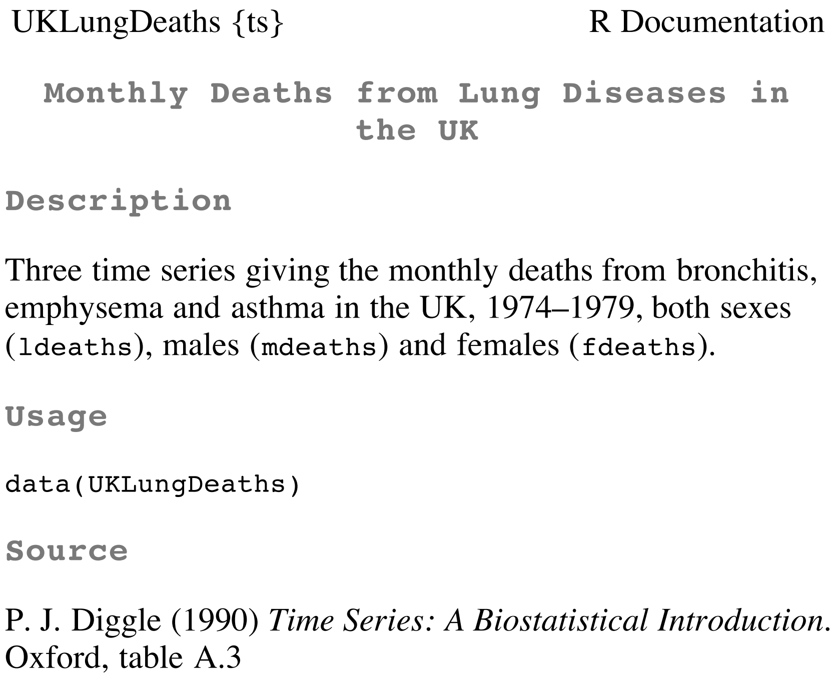
This exercise is intended to introduce you to some of the Time Series functions in R.

As you come across each new command, read the help file. While this may not help immediately, it will in the long term. Usually several readings are required before it all becomes more apparent!

To obtain the data for this session enter `R` and type `library(ts)`.

We will use the dataset `UKLungDeaths` so type `data(UKLungDeaths)`.

The `help(UKLungDeaths)` file in R is as follows:



UKLungDeaths {ts} R Documentation

Monthly Deaths from Lung Diseases in the UK

Description

Three time series giving the monthly deaths from bronchitis, emphysema and asthma in the UK, 1974–1979, both sexes (`ldeaths`), males (`mdeaths`) and females (`fdeaths`).

Usage

```
data(UKLungDeaths)
```

Source

P. J. Diggle (1990) *Time Series: A Biostatistical Introduction*. Oxford, table A.3

Set up the on-line help in R now.

1. Note that the data are given as a *time series structure*, not as a dataframe.

   We will consider the male death rates because there is a small trend present. Have a look at the data by typing `mdeaths`. Have a look at the two other time series.

   To investigate the time series structure, type

   ```
   class(mdeaths)
   attributes(mdeaths)
   ```

   There are several functions which extract aspects of the time base. Have a look at

   ```
   tsp(mdeaths)
   start(mdeaths)
   end(mdeaths)
   frequency(mdeaths)
   deltat(mdeaths)
   cycle(mdeaths)
   ```

2. There is a function `time` which takes a time series as its argument. Type in this command using `mdeaths` as its argument. What do you obtain?

3. The first thing to do with a time series is plot it. For a single series, you can type

   ```
   plot(mdeaths)
   ```

   There is also a special command `ts.plot()` for plotting a time series. Use this now to plot `mdeaths`.

   What is the clear message in the data, and why?

4. * `ts.plot` can plot several series simultaneously: try

   ```
     ts.plot(mdeaths, fdeaths, ldeaths)
   ```

   Obtain a better plot with

   ```
   ts.plot(mdeaths, fdeaths, ldeaths,
      gpars=list(xlab="year", ylab="deaths", lty=c(1:3)))
   ```

   Describe the main features of each time series.

   Add your name and a title, for example, by

   ```
    title("<your name> : deaths from lung diseases in the UK")
   ```

   and hand in your plot and description as part of Assignment 1.

5. Plot the male deaths data again but using points. This can be achieved by

   ```
       par(pch="*")
       ts.plot(mdeaths, type="p")
   ```

6. We shall smooth these data in various ways. Firstly, we use a simple moving average of order 3:

   ```
         m.ma3 <- filter(mdeaths, rep(1,3)/3)
   ```

   (Remember to keep reading the help files.)

   Add the smoothed series to the plot by

   ```
           lines(m.ma3, col="cyan")
   ```

7. * Repeat the above but with a moving average of order 13, as discussed in lectures. You could do this with

   ```
   m.ma13 <- filter(mdeaths, c(1/24, rep(1, 11)/12, 1/24))
   ```

   Plot the 13-point moving average and the three-point moving average on the same plot, comment on any interesting features that you observe, and hand in the plot and comments as part of Assignment 1.

   Now plot the 13-point moving average on its own: notice how important the scaling can be in interpretation. (Don't hand in this plot.)

8. Calculate the differences between the original series and the two smoothed versions and plot both sets of differences. This is how to find the differences (i.e., the residuals) from the three-point moving average:

   ```
   m.res3 <- mdeaths - m.ma3
   plot(m.res3)
   ```

   Now find and plot the residuals after subtracting the 13-point moving average.

   What do you see?

9. Finally in smoothing, we use the spline smoothing approach. Load the `modreg` library, then type

   ```
   males.ss <- smooth.spline(mdeaths)
   ```

   `smooth.spline` fits a cubic smoothing spline to the data. Again plot the series as points and add the fitted spline using `lines`. What does it resemble?

   Repeat the exercise for the three-point moving average residuals. There is a problem here because some of the time points do not have a smoothed value. To overcome this enter

   ```
   ma3.res <- window(m.res3,start=c(1974,2),end=c(1979,11))
   ma3.res <- as.ts(ma3.res)
   ```

   Now plot the residuals and fit the smoothing spline, adding the fit to the plot. What do you see?

   Note that `males.ss` and `res.ss` are not time series - you can check this by typing `is.ts(males.ss)`.

10. To remove a seasonal effect, we could take differences of lag 12:

    ```
    males.diff <- diff(mdeaths, lag=12)
    ```

    Plot the resultant series, and also the difference between this and the original series. What do you see?

11. For each time series found in this exercise (including the original `mdeaths`)

    ```
    acf(<name of series>)
    ```

    calculates the correlogram and plots it. Do this now. Comment on the shape for each series and think about what each plot indicates.

Questions 4 and 7 above are part of Assignment 1.

Assignment 1 is due by noon on Friday 27 August (end of Week 5).

Patty Solomon
July 2004