8

Long-Memory Processes

8.1 Purpose

Some time series exhibit marked correlations at high lags, and they are referred to as long-memory processes. Long-memory is a feature of many geophysical time series. Flows in the Nile River have correlations at high lags, and Hurst (1951) demonstrated that this affected the optimal design capacity of a dam. Mudelsee (2007) shows that long-memory is a hydrological property that can lead to prolonged drought or temporal clustering of extreme floods. At a rather different scale, Leland et al. (1993) found that Ethernet local area network (LAN) traffic appears to be statistically self-similar and a long-memory process. They showed that the nature of congestion produced by self-similar traffic differs drastically from that predicted by the traffic models used at that time. Mandelbrot and co-workers investigated the relationship between self-similarity and long term memory and played a leading role in establishing fractal geometry as a subject of study.

8.2 Fractional differencing

Beran (1994) describes the qualitative features of a typical sample path (realisation) from a long-memory process. There are relatively long periods during which the observations tend to stay at a high level and similar long periods during which observations tend to be at a low level. There may appear to be trends or cycles over short time periods, but these do not persist and the entire series looks stationary. A more objective criterion is that sample correlations r_k decay to zero at a rate that is approximately proportional to $k^{-\lambda}$ for some $0 < \lambda < 1$. This is noticeably slower than the rate of decay of r_k for realisations from an AR(1) process, for example, which is approximately proportional to λ^k for some $0 < \lambda < 1$.

The mathematical definition of a stationary process with long-memory, also known as long-range dependence or persistence, can be given in terms of

P.S.P. Cowpertwait and A.V. Metcalfe, *Introductory Time Series with R*, Use R, DOI 10.1007/978-0-387-88698-5_8,

its autocorrelation function. A stationary process x_t with long-memory has an autocorrelation function ρ_k that satisfies the condition

$$\lim_{k\to\infty} \rho_k = ck^{-\lambda}$$

for some $0 < c$ and $0 < \lambda < 1$. The closer λ is to 0, the more pronounced is the long-memory.

The hydrologist Harold Hurst found that for many geophysical records, including the Nile River data, a statistic known as the rescaled range (Exercise 4) over a period k is approximately proportional to k^H for some $H > \frac{1}{2}$. The Hurst parameter, H, is defined by $H = 1 - \lambda/2$ and hence ranges from $\frac{1}{2}$ to 1. The closer H is to 1, the more persistent the time series. If there is no long-memory effect, then $H = \frac{1}{2}$.

A fractionally differenced ARIMA process $\{x_t\}$, FARIMA(p, d, q), has the form

$$\phi(\mathbf{B})(1-\mathbf{B})^d x_t = \psi(\mathbf{B})w_t \tag{8.1}$$

for some $-\frac{1}{2} < d < \frac{1}{2}$. The range $0 < d < \frac{1}{2}$ gives long-memory processes. It can be useful to introduce the fractionally differenced series $\{y_t\}$ and express Equation (8.1) as

$$y_t = (1-\mathbf{B})^d x_t = [\phi(\mathbf{B})]^{-1}\psi(\mathbf{B})w_t \tag{8.2}$$

because this suggests a means of fitting a FARIMA model to time series. For a trial value of d, we calculate the fractionally differenced series $\{y_t\}$, fit an ARIMA model to $\{y_t\}$, and then investigate the residuals. The calculation of the fractionally differenced series $\{y_t\}$ follows from a formal binomial expansion of $(1-B)^d$ and is given by

$$(1-B)^d = 1 - dB + \frac{d(d-1)}{2!}B^2 - \frac{d(d-1)(d-2)}{3!}B^3 + \cdots$$

curtailed at some suitably large lag (L), which might reasonably be set to 40. For example, if $d = 0.45$, then

$$y_t = x_t - 0.450x_{t-1} - 0.12375x_{t-2} - 0.0639375x_{t-3} - \cdots - 0.001287312x_{t-40}$$

The R code for calculating the coefficients is

```
> cf <- rep(0,40)
> d <- 0.45
> cf[1] <- -d
> for (i in 1:39) cf[i+1] <- -cf[i] * (d-i) / (i+1)
```

Another equivalent expression for Equation (8.1), which is useful for simulations, is

$$x_t = [\phi(\mathbf{B})]^{-1}\psi(\mathbf{B})(1-\mathbf{B})^{-d}w_t$$

In simulations, the first step is to calculate $(1-B)^{-d}w_t$. The operator $(1-B)^{-d}$ needs to be expanded as

$$(1-B)^{-d} = 1 - d(-B) + \frac{-d(-d-1)}{2!}B^2 - \frac{-d(-d-1)(-d-2)}{3!}B^3 + \cdots$$

with the series curtailed at some suitably large lag L. The distributions for the independent white noise series can be chosen to fit the application, and in finance and telecommunications, heavy-tailed distributions are often appropriate. In particular, a t-distribution with ν (>4) degrees of freedom has kurtosis $6/(\nu-4)$ and so is heavy tailed. If, for example, $d = 0.45$ and $L = 40$, then

$$(1-B)^{-d}w_t = w_t + 0.45w_{t-1} + 0.32625w_{t-2} + 0.2664375w_{t-3} + \cdots + 0.0657056w_{t-40}$$

The autocorrelation function ρ_k of a FARIMA(0, d, 0) process tends towards

$$\frac{\Gamma(1-d)}{\Gamma(d)}|k|^{2d-1}$$

for large n. The process is stationary provided $-\frac{1}{2} < d < \frac{1}{2}$. This provides a relationship between the differencing parameter d and the long-memory parameter λ when $0 \le d$:

$$2d - 1 = -\lambda \iff d = \frac{1-\lambda}{2}$$

A FARIMA(0, d, 0) model, with $0 < d < \frac{1}{2}$, lies between a stationary AR(1) model and a non-stationary random walk. In practice, for fitting or simulation, we have to truncate a FARIMA(0, d, 0) process at some lag L. Then it is equivalent to an AR(L) model, but all the coefficients in the FARIMA(0, d, 0) model depend on the single parameter d.

8.3 Fitting to simulated data

In the following script, the function `fracdiff.sim` generates a realisation from a FARIMA process.[1] The first parameter is the length of the realisation, and then AR and MA parameters can be specified – use `c()` if there is more than one of each, followed by a value for d. The default for the discrete white noise (DWN) component is standard Gaussian, but this can be varied by using `innov` or `rand.gen`, as described in `help(fracdiff.sim)`. We then fit a FARIMA model to the realisation. In this case, we set the number of AR coefficients to be fitted to 1, but when fitting to a time series from an unknown model, we should try several values for the number of autoregressive and moving average parameters (`nar` and `nma`, respectively).

[1] You will need to have the `fracdiff` library installed. This can be downloaded from CRAN.

```
> library(fracdiff)
> set.seed(1)
> fds.sim <- fracdiff.sim(10000, ar = 0.9, d = 0.4)
> x <- fds.sim$series
> fds.fit <- fracdiff(x, nar = 1)
```

In the code below, the first `for` loop calculates the coefficients for the lagged terms in the fractional differences using the fitted value for d. The following nested loop then calculates the fractionally differenced time series. Then an AR model is fitted to the differenced series and the acf for the residuals is plotted (Fig. 8.2). The residuals should appear to be a realisation of DWN.

```
> n <- length(x)
> L <- 30
> d <- fds.fit$d
> fdc <- d
> fdc[1] <- fdc
> for (k in 2:L) fdc[k] <- fdc[k-1] * (d+1-k) / k
> y <- rep(0, L)
> for (i in (L+1):n) {
    csm <- x[i]
    for (j in 1:L) csm <- csm + ((-1)^j) * fdc[j] * x[i-j]
    y[i] <- csm
  }
> y <- y[(L+1):n]
> z.ar <- ar(y)
> ns <- 1 + z.ar$order
> z <- z.ar$res [ns:length(y)]
> par(mfcol = c(2, 2))
> plot(as.ts(x), ylab = "x")
> acf(x) ; acf(y) ; acf(z)
```

In Figure 8.1, we show the results when we generate a realisation $\{x_t\}$ from a fractional difference model with no AR or MA parameters, FARIMA(0, 0.4, 0). The very slow decay in both the acf and pacf indicates long-memory. The estimate of d is 0.3921. The fractionally differenced series, $\{y_t\}$, appears to be a realisation of DWN. If, instead of fitting a FARIMA(0, d, 0) model, we use `ar`, the order selected is 38. The residuals from AR(38) also appear to be a realisation from DWN, but the single-parameter FARIMA model is far more parsimonious.

In Figure 8.2, we show the results when we generate a realisation $\{x_t\}$ from a FARIMA(1, 0.4, 0) model with an AR parameter of 0.9. The estimates of d and the AR parameter, obtained from `fracdiff`, are 0.429 and 0.884, respectively. The estimate of the AR parameter made from the fractionally differenced series $\{y_t\}$ using `ar` is 0.887, and the slight difference is small by comparison with the estimated error and is of no practical importance. The residuals appear to be a realisation of DWN (Fig. 8.2).

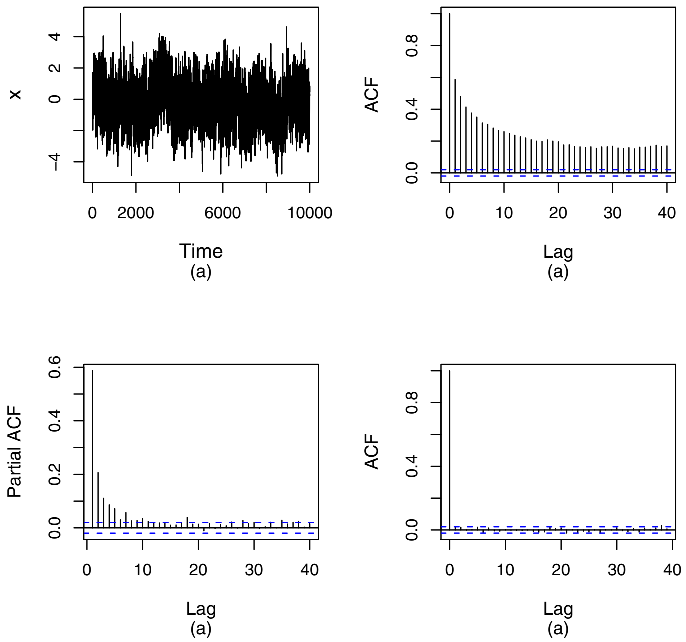

Fig. 8.1. A simulated series with long-memory FARIMA(0, 0.4, 0): (a) time series plot (x); (b) correlogram of series x; (c) partial correlogram of y; (d) correlogram after fractional differencing (z).

```
> summary(fds.fit)

...

Coefficients:
   Estimate Std. Error z value Pr(>|z|)
d   0.42904    0.01439    29.8   <2e-16 ***
ar  0.88368    0.00877   100.7   <2e-16 ***
ma  0.00000    0.01439     0.0        1
...

> ar(y)

Coefficients:
    1
0.887

Order selected 1  sigma^2 estimated as  1.03
```

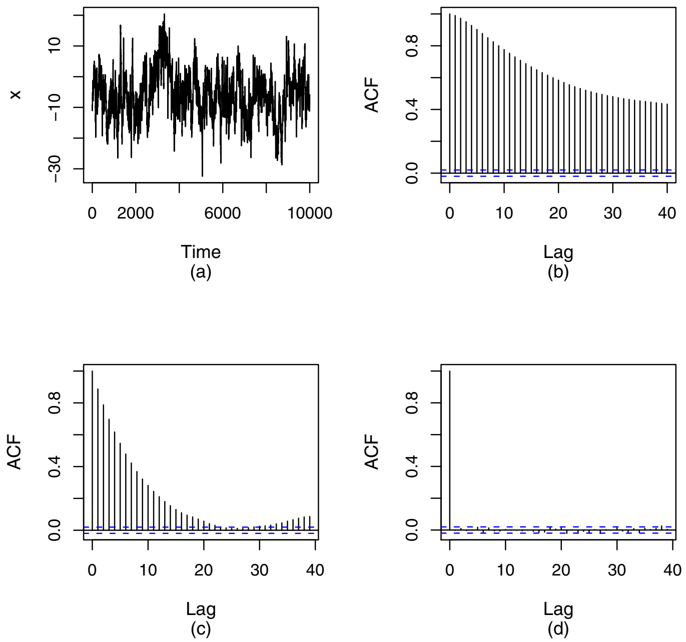

Fig. 8.2. A time series with long-memory FARIMA(1, 0.4, 0): (a) time series plot (`x`); (b) correlogram of series `x`; (c) correlogram of the differenced series (`y`); (d) correlogram of the residuals after fitting an AR(1) model (`z`).

8.4 Assessing evidence of long-term dependence

8.4.1 Nile minima

The data in the file `Nilemin.txt` are annual minimum water levels (mm) of the Nile River for the years 622 to 1284, measured at the Roda Island gauge near Cairo. It is likely that there may be a trend over a 600-year period due to changing climatic conditions or changes to the channels around Roda Island. We start the analysis by estimating and removing a linear trend fitted by regression. Having done this, a choice of `nar` is taken as a starting value for using `fracdiff` on the residuals from the regression. Given the iterative nature of the fitting process, the choice of initial values for `nar` and `nma` should not be critical. The estimate of d with `nar` set at 5 is 0.3457. The best-fitting model to the fractionally differenced series is AR(1) with parameter 0.14. We now re-estimate d using `fracdiff` with `nar` equal to 1, but in this case the estimate of d is unchanged. The residuals are a plausible realisation of DWN. The acf of the squared residuals indicates that a GARCH model would be appropriate. There is convincing evidence of long-term memory in the Nile River minima flows (Fig. 8.3).

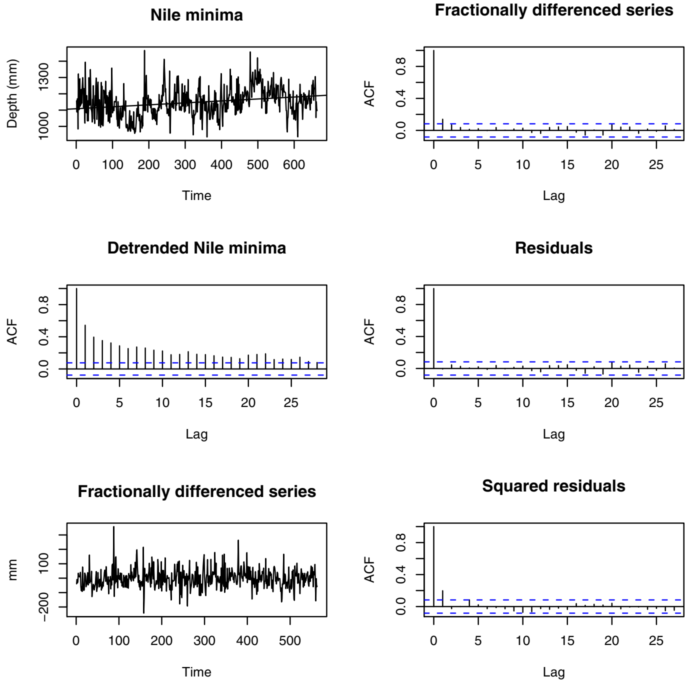

Fig. 8.3. Nile River minimum water levels: time series (top left); acf of detrended time series (middle left); fractionally differenced detrended series (lower left); acf of fractionally differenced series (top right); acf of residuals of AR(1) fitted to fractionally differenced series (middle right); acf of squared residuals of AR(1) (lower right).

8.4.2 Bellcore Ethernet data

The data in `LAN.txt` are the numbers of packet arrivals (bits) in 4000 consecutive 10-ms intervals seen on an Ethernet at the Bellcore Morristown Research and Engineering facility. A histogram of the numbers of bits is remarkably skewed, so we work with the logarithm of one plus the number of bits. The addition of 1 is needed because there are many intervals in which no packets arrive. The correlogram of this transformed time series suggests that a FARIMA model may be suitable.

The estimate of d, with `nar` set at 48, is 0.3405, and the fractionally differenced series has no substantial correlations. Nevertheless, the function `ar` fits an AR(26) model to this series, and the estimate of the standard deviation of the errors, 2.10, is slightly less than the standard deviation of the fractionally differenced series, 2.13. There is noticeable autocorrelation in the series of squared residuals from the AR(26) model, which is a feature of time series that have bursts of activity, and this can be modelled as a GARCH

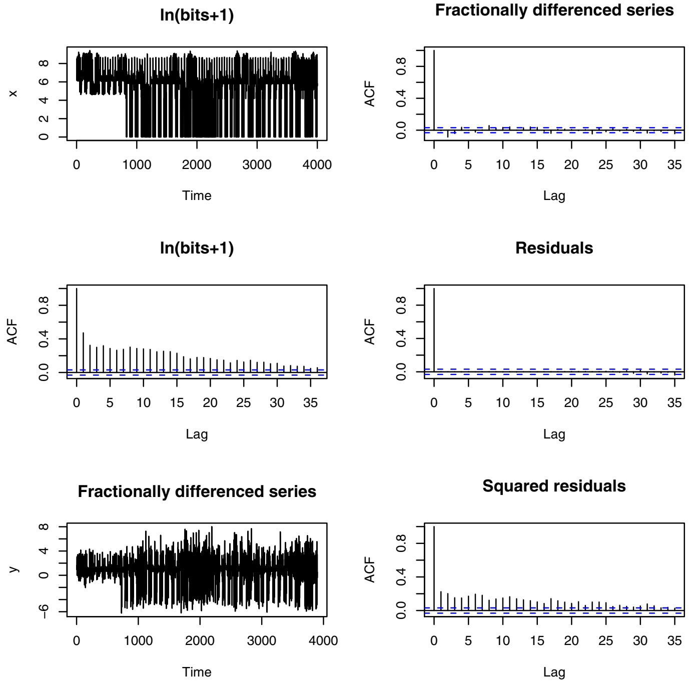

Fig. 8.4. Bellcore local area network (LAN) traffic, ln(1+number of bits): time series (top left); acf of time series (middle left); fractionally differenced series (lower left); acf of fractionally differenced series (top right); acf of residuals of AR(26) fitted to fractionally differenced series (middle right); acf of squared residuals of AR(26) (lower right).

process (Fig. 8.4). In Exercises 1 and 2, you are asked to look at this case in more detail and, in particular, investigate whether an ARMA model is more parsimonious.

8.4.3 Bank loan rate

The data in `mprime.txt` are of the monthly percentage US Federal Reserve Bank prime loan rate,[2] courtesy of the Board of Governors of the Federal Reserve System, from January 1949 until November 2007. The time series is plotted in the top left of Figure 8.5 and looks as though it could be a realisation of a random walk. It also has a period of high variability. The correlogram shows very high correlations at smaller lags and substantial correlation up to lag 28. Neither a random walk nor a trend is a suitable model for long-term

[2] Data downloaded from Federal Reserve Economic Data at the Federal Reserve Bank of St. Louis.

simulation of interest rates in a stable economy. Instead, we fit a FARIMA model, which has the advantage of being stationary.

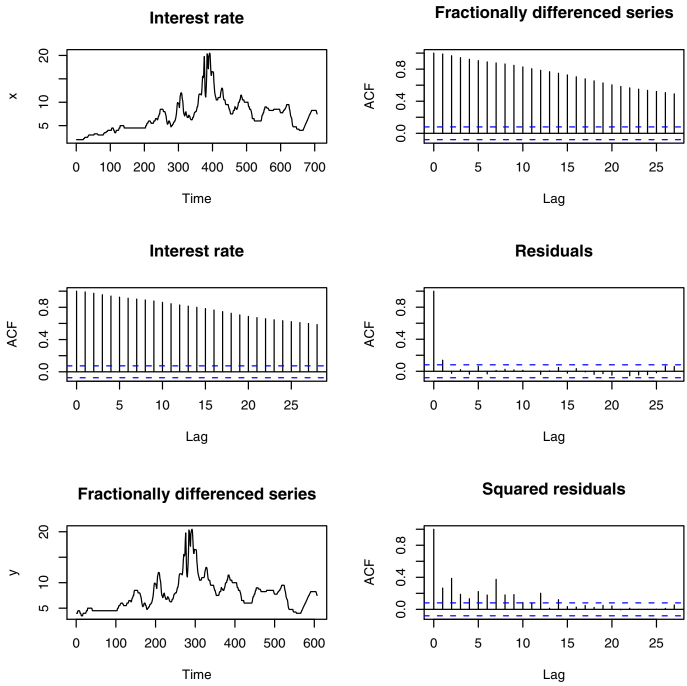

Fig. 8.5. Federal Reserve Bank interest rates: time series (top left); acf of time series (middle left); fractionally differenced series (lower left); acf of fractionally differenced series (upper right); acf of residuals of AR(17) fitted to fractionally differenced series (middle right); acf of squared residuals of AR(17) (lower right).

The estimate of d is almost 0, and this implies that the decay of the correlations from an initial high value is more rapid than it would be for a FARIMA model. The fitted AR model has an order of 17 and is not entirely satisfactory because of the statistically significant autocorrelation at lag 1 in the residual series. You are asked to do better in Exercise 3. The substantial autocorrelations of the squared residuals from the AR(17) model indicate that a GARCH model is needed. This has been a common feature of all three time series considered in this section.

8.5 Simulation

FARIMA models are important for simulation because short-memory models, which ignore evidence of long-memory, can lead to serious overestimation of system performance. This has been demonstrated convincingly at scales from reservoirs to routers in telecommunication networks.

Realistic models for simulation will typically need to incorporate GARCH and heavy-tailed distributions for the basic white noise series. The procedure is to fit a GARCH model to the residuals from the AR model fitted to the fractionally differenced series. Then the residuals from the GARCH model are calculated and a suitable probability distribution can be fitted to these residuals (Exercise 5). Having fitted the models, the simulation proceeds by generating random numbers from the fitted probability model fitted to the GARCH residuals.

8.6 Summary of additional commands used

`fracdiff`	fits a fractionally differenced, FARIMA(p, d, q), model
`fracdiff.sim`	simulates a FARIMA model

8.7 Exercises

1. Read the LAN data into `R`.
   a) Plot a boxplot and histogram of the number of bits.
   b) Calculate the skewness and kurtosis of the number of bits.
   c) Repeat (a) and (b) for the logarithm of 1 plus the number of bits.
   d) Repeat (a) for the residuals after fitting an AR model to the fractionally differenced series.
   e) Fit an ARMA(p, q) model to the fractionally differenced series. Is this an improvement on the AR(p) model?
   f) In the text, we set `nar` in `fracdiff` at 48. Repeat the analysis with `nar` equal to 2.

2. Read the LAN data into `R`.
   a) Calculate the number of bits in 20-ms intervals, and repeat the analysis using this time series.
   b) Calculate the number of bits in 40-ms intervals, and repeat the analysis using this time series.
   c) Repeat (a) and (b) for realisations from FARIMA(0, d, 0).

3. Read the Federal Reserve Bank data into `R`.
   a) Fit a random walk model and comment.
   b) Fit an ARMA(p, q) model and comment.

4. The rescaled adjusted range is calculated for a time series $\{x_t\}$ of length m as follows. First compute the mean, $\bar{x}$, and standard deviation, s, of the series. Then calculate the adjusted partial sums

$$S_k = \sum_{t=1}^{k} x_t - k\bar{x}$$

for $k = 1, \ldots, m$. Notice that $S(m)$ must equal zero and that large deviations from 0 are indicative of persistence. The rescaled adjusted range

$$R_m = \{\texttt{max}(S_1, \ldots, S_m) - \texttt{min}(S_1, \ldots, S_m)\}/s$$

is the difference between the largest surplus and the greatest deficit. If we have a long time series of length n, we can calculate R_m for values of m from, for example, 20 upwards to n in steps of 10. When m is less than n, we can calculate $n - m$ values for R_m by starting at different points in the series. Hurst plotted $\ln(R_m)$ against $\ln(m)$ for many long time series. He noticed that lines fitted through the points were usually steeper for geophysical series, such as streamflow, than for realisations of independent Gaussian variables (Gaussian DWN). The average value of the slope (H) of these lines for the geophysical time series was 0.73, significantly higher than the average slope of 0.5 for the independent sequences. The linear logarithmic relationship is equivalent to

$$R_m \propto m^H$$

Plot $\ln(R_m)$ against $\ln(m)$ for the detrended Nile River minimum flows.

5. a) Refer to the data in `LAN.txt` and the time series of logarithms of the numbers of packet arrivals, with 1 added, in 10-ms intervals calculated from the numbers of packet arrivals. Fit a GARCH model to the residuals from the AR(26) model fitted to the fractionally differenced time series.
   b) Calculate the residuals from the GARCH model, and fit a suitable distribution to these residuals.
   c) Calculate the mean number of packets arriving in 10-ms intervals. Set up a simulation model for a router that has a realisation of the model in (a) as input and can send out packets at a constant rate equal to the product of the mean number of packets arriving in 10-ms intervals with a factor g, which is greater than 1.
   d) Code the model fitted in (a) so that it will provide simulations of time series of the number of packets that are the input to the router. Remember that you first obtain a realisation for ln(number of packets + 1) and then take the exponential of this quantity, subtract 1, and round the result to the nearest integer.

e) Compare the results of your simulation with a model that assumes Gaussian white noise for the residuals of the AR(26) model for $g =$ 1.05, 1.1, 1.5, and 2.