## 2.9 Long Memory models and Fractional Differencing

Recent statistical literature has concerned with a study of long memory models which go beyond the presence of random walks and unit roots in the univariate time series processes. Following Mandelbrot and Van Ness (1968) fractional Brownian motion has been applied to “strongly dependent" geophysical phenomenon, McLeod and Hipel (1978). The objectives are threefold: (1) Explicitly account for persistence, (2) Incorporate both short and long term correlations in the data, and (3) Aid in forecasting. Hosking (1981) suggests that these three objectives can be attained with fractional differencing. Apparently unaware of these developments, Campbell and Mankiw (1987) suggest that economists should focus attention not merely on the existence of unit roots, but also consider their “quantitative importance." Similarly, the expressions “big" random walk in GNP used by Cochrane (1988), or “mean reversion" used by Poterba and Summers (1988) are designed to attract the attention of economists to a study of long-term persistence. Low power of the asymptotic unit root type tests is noted by Sowell (1990), Lo and MacKinlay (1989) and Diebold and Rudebusch (1991).

Autoregressive fractionally integrated moving average, ARFIMA(p,d,q), processes are known to be capable of modeling long-run persistence. Introduced by Granger and Joyeux (1980), they generalize Box-Jenkins models, when the order d of differencing (or integration) is allowed to be fractional. Geweke and Porter-Hudak (GPH, 1983) suggest a frequency domain regression to estimate d. As shown in Vinod (1993) GPH has serious shortcomings. A maximum likelihood ML estimation method is implemented in the R package called “fracdiff.” based on Haslett and Raftery (1989) algorithm in *Applied Statistics*. Brockwell and Davis (1987, Sec. 12.4) discuss the theory in suitable detail.

The general expression for ARFIMA models is an extension of the expression in (2.4.1) upon excluding the seasonal part, but allowing d to be fractional. ARFIMA(p,d,q) is

$$(1-L)^d y_t = \mu + \frac{\theta(L)}{\phi(L)} a_t \qquad (2.9.1)$$

where $y_t$ is the original data. As before, polynomial representing coefficients of the MA part are in the numerator while the AR part polynomial is in the denominator. In writing the polynomials in a ratio form we remove any common roots and assume that the polynomials represent convergent series. That is, we have well behaved roots of the characteristic polynomials θ and ϕ. The white noise input is a zero mean white noise process with variance $\sigma^2$

$$a_t \sim WN(0, \sigma^2) \qquad (2.9.2)$$

The new and distinguishing feature is that d is fractional, usually with $|d|<0.5$.The ARFIMA where d is a fraction relies on the binomial expansion valid for any $d > -1$:

$$\Delta^d = (1-L)^d = 1-d\,L - (1/2!)d(1-d)L^2 - (1/3!)d(1-d)(2-d)L^3 \ldots, \qquad (2.9.3)$$

using the factorial notation r! for a product of numbers from 1 to r. The theory uses the gamma function relation $\Gamma(d)=d^{-1}\Gamma(d+1)$, by analytic continuation. Even though $|d|<1$ are all fractional, the choice $|d|<0.5$ is of practical interest, since it can be shown that the variance does not diverge for that choice.

If $y_t \sim$ ARFIMA(0,d,0), it can be expressed as MA($\infty$) process. The j-th coefficient of MA($\infty$) representation with coefficients, $\psi_j = \Pi_{0<k\leq j}\,[(k-1+d) / k]$. Hence it can be shown that the autocorrelation function is

$$\rho_k = \Pi_{0<j\leq k}\,[(j-1+d) / (j-d)], \tag{2.9.4}$$

which is approximately written as

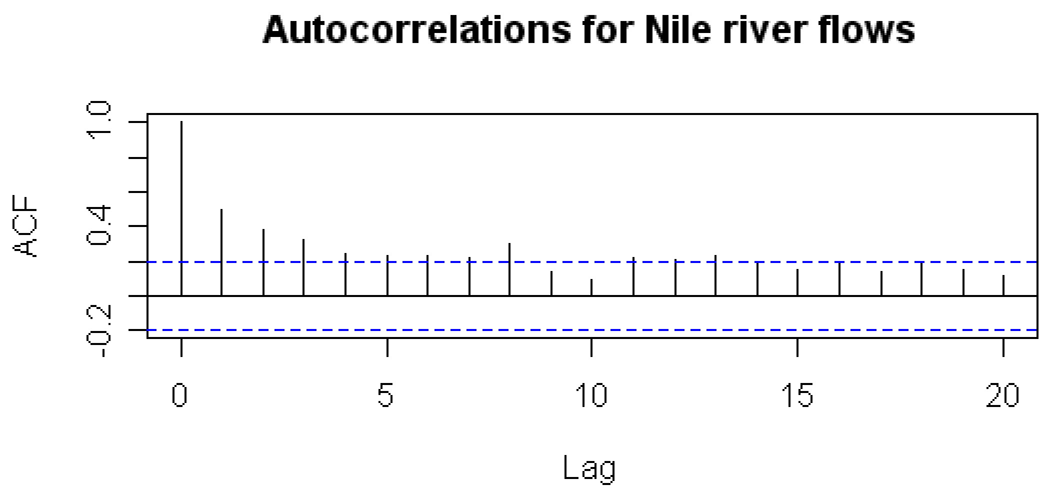
$$\rho_k \approx k^{2d-1}\Gamma(1-d) / \Gamma(d), \text{ as } k \to \infty \tag{2.9.5}$$

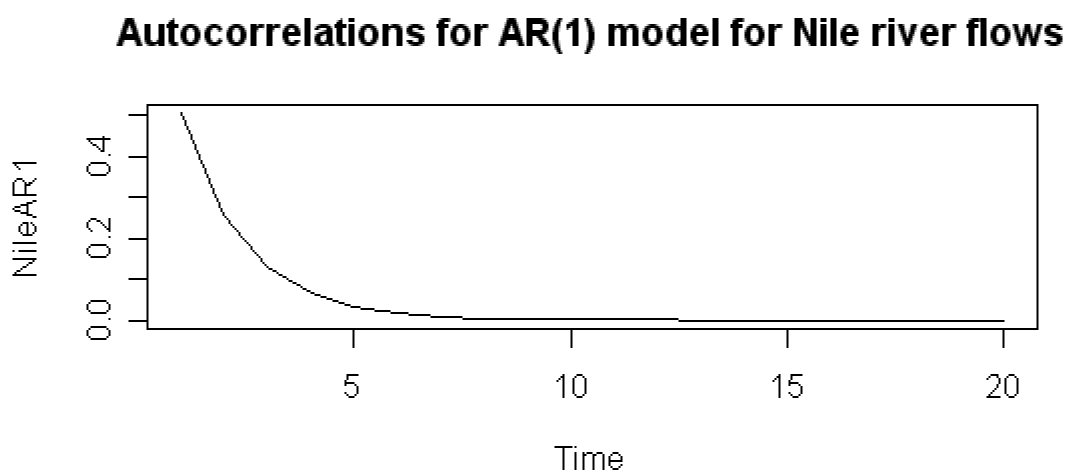
Recall from Section 2.4 that the autocorrelations of order k, $\rho_k = \phi\ \rho_{k-1} = (\phi^{\,k})\rho_0$ decline (exponentially) fast toward zero for $|\phi|<1$ as the power k increases for the AR(1) process. By contrast, ARFIMA process has *long memory* in the sense that autocorrelations at very long lags are nonzero. The slow decline arises because the $\rho_k$ for ARFIMA is proportional to $k^{2d-1}$ in (2.9.5).

The power index $2d-1$ of k, which is much larger than unity ($k>>1$) is worth considering. If $|d|<0.5$, $2d-1<0$ and we are raising $k>>1$ to a negative power, allowing it to become zero. If $d=0.5$, $2d-1$ becomes zero and $\rho_k \to \Gamma(1-d)/\Gamma(d)=1$, as $k \to \infty$. Since near-perfect autocorrelation at very large lag seems unrealistic, $d=0.5$ is generally considered unrealistic also. If $0.5<d<1$, the index $2d-1$ becomes positive power for $k>>1$ leading to unstable and unrealistic process with infinite variance. In summary, the dependence between observations decays exponentially fast for the usual ARMA models, whereas for ARFIMA it decays hyperbolically slow depending on the value of the fractional parameter d, with $|d|<0.5$ generally held to be plausible. Vinod (1993) argues that d is a powerful tool for summarizing the behavior of a time series.

Long memories may be crucial in some applications. Historically, the interest in long memory series was prompted by the observation that the flow of great rivers like the Nile in Egypt is slow and smooth. Measurements of the annual flow of the river Nile at Ashwan 1871–1970 are always available in R. Consider a snippet.

```
#R2.9.1
arima(Nile, order=c(1,0,0)) #yields AR(1) coefficient of 0.5063
kk=1:20; NileAR1=0.5063^kk; plot.ts(NileAR1)
par(mfrow=c(2,1))
acf(Nile, main="Autocorrelations for Nile river flows")
plot.ts(NileAR1, main="Autocorrelations for AR(1) model for Nile river
flows")
```




The figures show clearly that a long-memory model is more appropriate for Nile river flows than AR(1), since the autocorrelations do indeed decline slowly. Now we turn to estimation of fractional differencing models in R. The R package **fracdiff** by V. A. Reisen gives details. We only give brief set of results for the Nile data and US GNP data.

```
#R2.9.2
library(fracdiff)
fdGPH(Nile, bandw.exp = 0.5)#gives the GPH estimate d= 0.3896247
fr1=fracdiff(Nile)
summary(fr1)#gives the ML estimate d=0.3639127
confint(fr1) #prints 95% confidence interval (0.3638995 0.3639258)

library(tseries)
data(USeconomic)
summary(USeconomic)
dgnp=diff(log(GNP),1)
```

```
fracdiff(log(GNP))  # the d is too close to 0.5 an unstable level
dgnp=diff(log(GNP),1)#compute first difference
fracdiff(dgnp)# this d =0.2233329 is more plausible
```

# 2.10 Forecasting

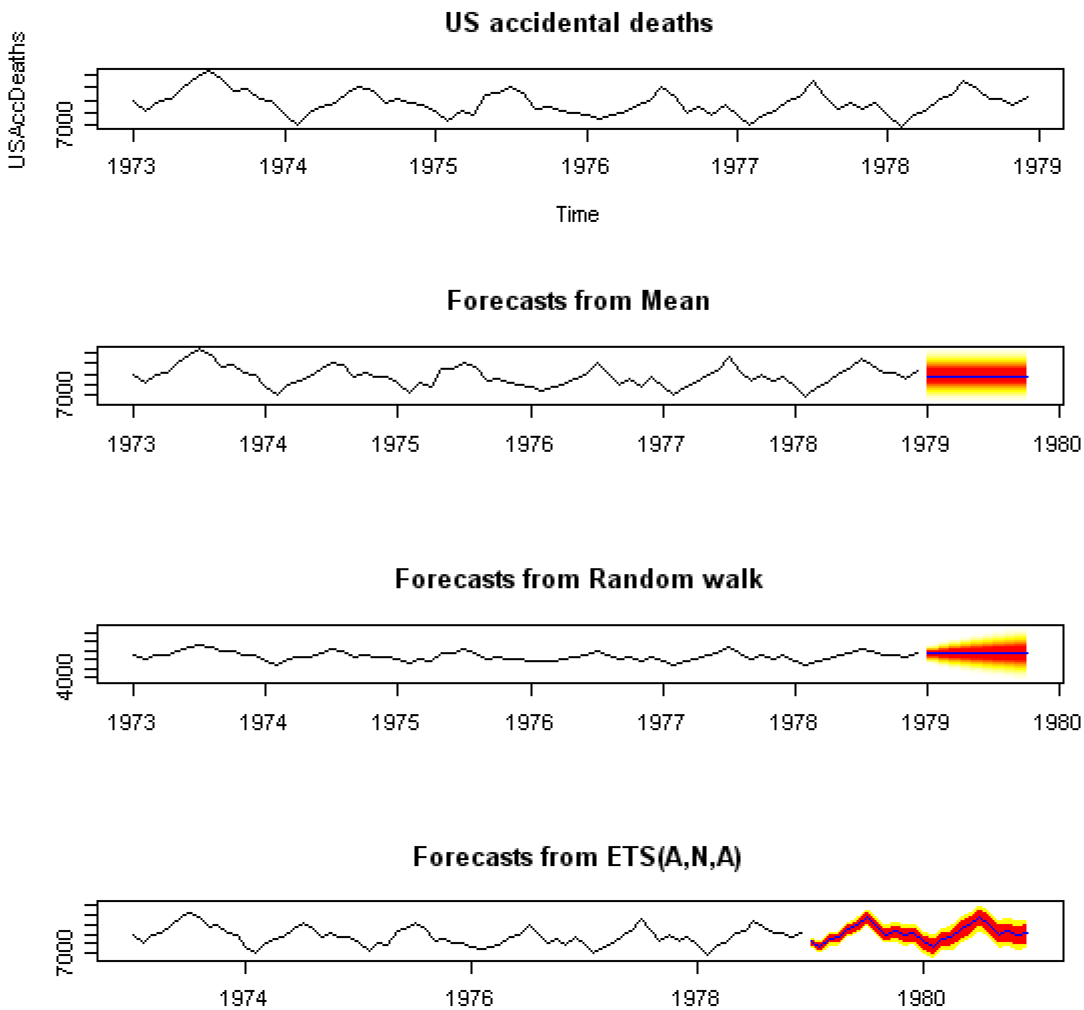
Once a suitable model is fitted for a series, it is a simple matter to use it for forecasting. For example, if $y=b_1+b_2x_1+b_3x_2$ is a fitted model and we want to find y for a given set of future values of $x_{1,0}$ and $x_{2,0}$ then the forecast y is simply $b_1+b_2x_{1,0}+b_3x_{2,0}$. Although in the univariate setup of this chapter, $x_1$ and $x_2$ might be the lagged values of y, the forecasting from a model is not so restricted. The right hand variables, in general, will be other regressors.

The R software is particularly suitable for this task, as it imposes little additional work. All fitted coefficients similar to $b_1$, $b_2$ and $b_3$ are readily accessible from the regression object used for fitting, without having to type any of their values. The R command "predict" actually gets fitted values not predictions. The reader should consult the help facility in R for details of several cousins of predict including: predict.lm, predict.loess, predict.nls, predict.poly, predict.princomp, predict.smooth.spline, predict.ar, predict.Arima, predict.arima0, predict.HoltWinters, and predict.StructTS. The following snippet allows one to set the number of periods ahead.

```
#R2.10.1  EXAMPLE ARIMA(p,d,q)x(P,D,Q)for seasonal
a1=arima(AirPassengers, order = c(1, 1, 0), seasonal = list(order =
c(2, 1, 0), period = 12))
predict(a1, n.ahead=6)
```

The R package called forecast (part of package forecasting) offers several further facilities for forecasting and plotting. For example, consider forecasting of US accidental deaths by some methods and plotting of forecasts with confidence intervals. For illustration, we report only a few methods among several available in the package: (i) simple mean based forecasting (ii) random walk forecasting (iii) forecasting using a state space exponential smoothing model by Hyndman et al (2002) is given by:

```
#R2.10.1  EXAMPLE of Forecasting US accidental deaths
library(forecast)
par(mfrow = c(4, 1)) #set up 4 plots in one
plot(USAccDeaths, main="US accidental deaths") #plot original series
accd.fcast <- meanf(USAccDeaths, h=10, fan=T)#h=number of periods
plot(accd.fcast)
# random walk forecast rwf
plot(rwf(USAccDeaths, h=10, drift=FALSE, conf=c(80,95), fan=T))
fit <- ets(USAccDeaths)#exponential smoothing state space model
plot(forecast(fit))
```




Importance of Out-of-sample Forecasting: Given data with T points, the performance of an empirical model explaining the data is often measured by the relative "smallness, unbiasedness," and other randomness properties of model errors approximated by observable residuals. There are purely statistical tools, which can mechanically improve the goodness of "in-sample" fit. In some cases even a near-perfect fit can be achieved. Yet, it is well known that "in-sample" fit is only an important first step. A model that performs near-perfect in sample, may be poor out-of-sample and may not have any reason to work from known relations among economic agents. Economists have an unspoken preference for models with sound "theory," judged by the mathematical complexity, no matter the assumptions are unverifiable. Since in-sample performance says nothing about theoretical suitability of a model, near-perfect fit by itself will not impress almost anyone in Economics.

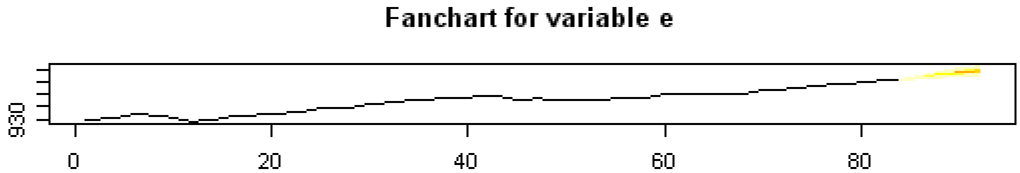
How to study out-of-sample performance of a model? Given a long enough time series, it is a simple matter to split its length into two components, keeping the first T observations (say) for in-sample and reserving the last k (say) for out-of-sample assessment. Average squared error out-of-sample is a common criterion for comparing models. Theil has developed a range of criteria.

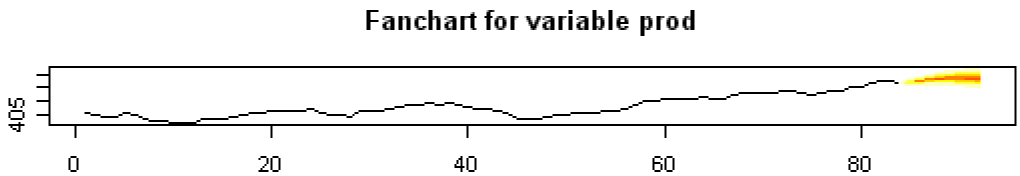
REMARK 2.10.1: Ideally the data retained for out-of-sample assessment should be representative of the entire data. For example, if data refer to stock prices and the k observations retained for out-of-sample are all belonging to the boom period, the assessment will be biased. There is an old school of thought in Economics, which argues that ranking and suitability of economic models should be based solely on their out-of-sample forecasts. Unfortunately, it is a whole new unsolved problem to make sure that the set of k observations retained for out-of-sample testing is large enough and representative enough to help decide whether the model will work for the unknown future.

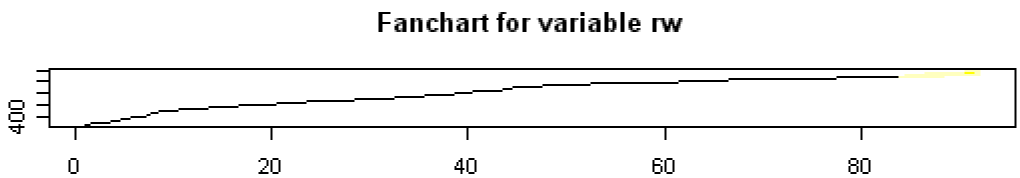
Clearly, forecasting is an important topic, which will be mentioned in various subsequent chapters dealing with more variables than one. These models have to address the separate issue of forecasting the right hand side variables before a forecast for the left hand side variable can be made. Often the univariate models of this chapter are used for forecasting the right side variables.

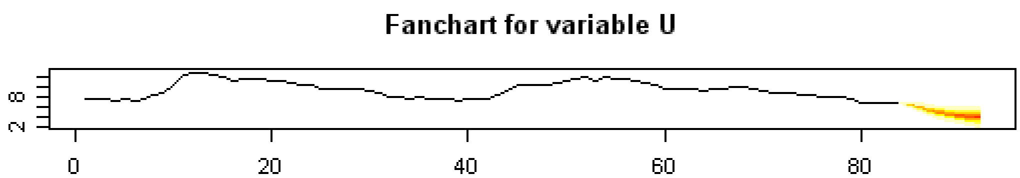
<u>Multivariate Forecasting:</u> We provide in passing some vector autoregression (VAR) multivariate model forecasting using R package called ‘vars,’ even though this chapter is not concerned with multivariate models. We provide forecasting charts using VAR models for employment, labor productivity, real wages and unemployment rate. A 95% forecast interval is also plotted for 8-step ahead forecasting exercise, that is over next 2 years or 8 quarters.

```
library(vars); data(Canada)#load macroeconomic data for Canada
par(mfrow = c(4, 1)) #set up 4 plots in one
#prod= labour productivity; e=employment; U=unemployment rate
# rw = real wage  VAR vector autoregression estimated as
var.2c <- VAR(Canada, p = 2, type = "const")
var.2c.prd <- predict(var.2c, n.ahead = 8, ci = 0.95)
fanchart(var.2c.prd) # does beautiful fanplot for forecasting
```




# 2.11 Solved Examples for Chapter Review

Ex.1] If y is a short time series with values: 10, 8, 22, 17, 21, 28, 10, 23, 24, 9. Fit an AR(2) model and state if it can represent some kind of cyclical behavior.

Answer: Recall that autoregressions can be fitted by ordinary least squares. Only when the ARIMA(p,d,q) models have MA components ($q \neq 0$), we absolutely need maximum likelihood nonlinear methods implemented by the function arima in R. Recall from (2.3.2) to (2.3.5) that the roots of the characteristic polynomial involve the radical, $R_{ad}$= =$[(\phi_1)^2 + 4\,\phi_2]$. We need to evaluate $R_{ad}$= for our data. If $R_{ad} > 0$ the roots are real and the dynamics do not indicate cyclical behavior. If on the other hand, $R_{ad} < 0$, the square root involves imaginary numbers and they alone create a cyclical behavior. The following snippet is one way of doing this.

```
#R2.11.1
```

```
y=c(10,  8, 22, 17, 21, 28, 10, 23, 24,  9)
a2=arima(y, order = c(2, 0, 0))
#AR order p=2, differencing order d=0, and MA order q=0
summary(a2)
a2$coef[1:2]
phi=as.numeric(a2$coef[1:2])
rad=phi[1]^2 + 4*phi[2]
rad  # -1.474175, is negative implies cyclical behavior
```

Ex.2. Use the co2 data always available in R to decompose (seasonally) it and fit a linear trend to its trend part. What is the slope of the trend? Does it indicate global warming?

Answer: We use either the decompose function or sophisticated stl function in R. The latter needs a specification of the seasonal span called s.window. Since the data are monthly, we can choose 12 as the span. We first create an object containing the decomposition called sc. The plots of various components are produced and trend part is extracted by attaching $trend to the object. Next create tim as a regressor and fit a linear trend. Positive trend suggests global warming. The R program to do this is:

```
#R2.11.2
sc=stl(co2, s.window=12)
plot(sc)
y=sc$time.series[,"trend"]; n=length(y); tim=1:n; reg1=lm(y~tim)
reg1
#(Intercept)          tim
#   311.4468       0.1092
```

Decomposition of CO2 levels