- paragraphs run together
- ALL displays too small
- PROOF READ before you submit something!!! Don't make me do it for you.

# Time Series Analysis with R

A. Ian McLeod, Hao Yu, Esam Mahdi

*Department of Statistical and Actuarial Sciences, The University of Western Ontario, London, Ont., Canada N6A 5B7*

## 1. Introduction

In the area of computational time series analysis, especially for advanced algorithms, it has become clear that the R has established itself as the choice of many researchers. R is widely used not only by researchers but also in diverse applications and teaching. Naturally, there are many other software systems such as *Mathematica* (Wolfram Research, 2011), that have more capabilities such as curated databases and symbolic computation (Zhang and McLeod, 2006) that may be useful in some time series research. For most researchers working with time series, R provides an excellent broad platform. This history of R has been discussed elsewhere (Gentleman and Ihaka, 1996) so before continuing our survey we will just point out some other key features of this quantitative programming environment (QPE). R incorporates many years of previous research in statistical and numerical computing and so it is built on a solid foundation of core statistical and numerical algorithms. The R programming language is a functional, high-level interactive and scripting language that offers two levels of object-oriented programming. For an experienced R user, using this language to express an algorithm is often easier than using ordinary mathematical notation and it is much more powerful since, unlike mathematical notation, it can be evaluated. In this way, R is an important tool of thought. Novice and casual users of R may interact with it using Microsoft Excel (Heiberger and Neuwirth, 2009) or R Commander (Fox, 2005). Through the use of Sweave (Leisch, 2002, 2003), R supports high-quality technical typesetting and reproducible research including

*Email addresses:* aim@stats.uwo.ca (A. Ian McLeod), hyu@stats.uwo.ca (Hao Yu), emahdi@uwo.ca (Esam Mahdi)

reproducible applied statistical and econometric analysis (Kleiber and Zeileis, 2008). This article has been prepared using Sweave and R scripts for all computations including all figures and tables are available in an online supplement. R supports 64-bit, multicore, parallel and cluster computing (Schmidberger et al., 2009; Hoffmann, 2011; Revolution Computing, 2011). There are a number of reasons why R has recently attained a dominant position in statistics and time series in the areas of research, teaching and applications. Certainly an important factor has been that it is an open source project, providing a freely available and high quality computing environment with thousands of add-on packages. There is a vast literature available on R that includes introductory books as well as treatments of specialized topics. General purpose introductions to R are available in many books (Braun and Murdoch, 2008; Crawley, 2007; Dalgaard, 2008; Adler, 2009; Everitt and Hothorn, 2009; Zuur et al., 2009) as well as the freely available one Venables et al. (2011). Advanced aspects of the R programming are treated by (Venables and Ripley, 2000; Spector, 2008; Chambers, 2008; Gentleman, 2009). As of April 2011, Springer has published 28 titles in the *Use R* book series and Chapman & Hall/CRC has many forthcoming titles in *The R Series*. There are many other high quality books that feature R. Many of these books discuss R packages developed by the author of the book and others provide a survey of R tools useful in some application area. In addition to this flood of high quality books, the *Journal of Statistical Software* (JSS) publishes refereed papers discussing statistical software. JSS reviews not only the paper but the quality computer code as well and publishes both the paper and code on its website. Many of these papers are subsequently available on the CRAN, the Comprehensive R network `http://cran.r-project.org/`. The rigorous review process ensures a high quality standard. For this reason, in this article, the focus will be on R packages that are accompanied by published books and papers in JSS. There is also the specialized refereed journal, *The R Journal*. Rmetrics (Würtz, 2004) provides R packages for teaching and research in quantitative finance and time series analysis that are further described in the electronic books that they publish. There are numerous textbooks, suitable for a variety of courses in time series analysis (Venables and Ripley, 2002; Chan, 2010; Cryer and Chan, 2008; Lütkepohl and Krätzig, 2004; Shumway and Stoffer, 2011; Tsay, 2010). These books incorporate R usage in the book and provide an R package on CRAN that includes scripts and datasets used in the book.

This paragraph just about says it correctly. If a reader has never used R, this article may get them to consider using it, but I do not think there is enough in here to actually get them started using R. Someone who knows R most likely knows most of the material in this paper, or can use the task view to get help. There is a chance that an experienced user might find a command s/he hasn't seen yet.

The purpose of our article is to provide an overview of some of the published high-quality computational time series research using R. If you are not already an R user, this article may help you in learning about R. Existing R may find the brief overview of existing R software for time series analysis of interest.

Existing R may find ??

## 2. Time series plots

In this section our focus is on plots of the time series. Such plots are often the first step in an exploratory analysis and are usually provided in a final report. R can produce a variety of these plots not only for regular time series but also for more specialized time series such as irregularly-spaced time series. The built-in function `plot()` may be used to plot simple series such as the annual lynx series, `lynx`. The aspect-ratio is often helpful in visualizing slope changes in a time series (Cleveland et al., 1988; Cleveland, 1993) For many time series an aspect-ratio of 1/4 is good choice. The function `xyplot()` (Sarkar, 2008) allows one to easily control the aspect-ratio. Figure 1 shows the time series plot of the lynx series with an aspect-ratio of 0.25. The asymmetric rise and fall of the population is easily noticed with the choice of the aspect-ratio. There are many possible styles

can't see these very well

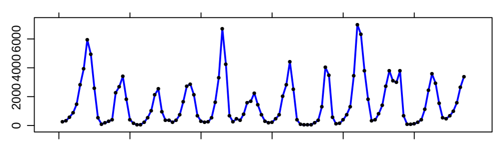


Figure 1: Annual numbers of lynx trappings in Canada.

for time series plots. Sometimes a high-density line plot is effective as in Figure 2.

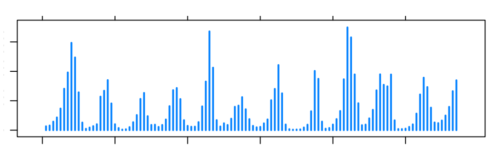


Figure 2: High density line plot.

Another capability of `xyplot()` is the cut-and-stack time series plot for longer series. Figure 3 shows a cut-and-stack plot of the famous Beveridge wheat price index using `xyplot()` and `asTheEconomist()`. The cut-and-stack plot uses the equal-count-algorithm (Cleveland, 1993, p. 134) to divide the series into a specified number of subseries using a 50% overlap.

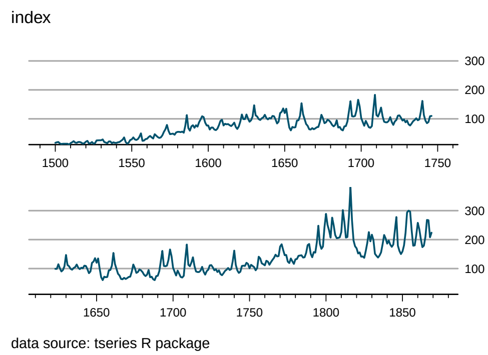


Figure 3: Beveridge wheat price index.

Figure 4 uses `xyplot()` to plot the seasonal decomposition of the well-known $CO_2$ time series. The seasonal adjustment algorithm available in R `stl()` is described in the R function documentation and in more detail Cleveland et al. (1990); Cleveland (1993) This plot efficiently reveals a large amount of information. For example, Figure 4, reveals that the seasonal amplitudes are increasing.

Bivariate or multivariate time series may also be plotted with `xyplot()`. In Figure 5, time series plot for the annual temperature in F° for Canada, England and China 1970-2007, is shown. The regular plot function, `plot()` would plot these series superimposed on a single graph but the juxtaposed version is preferable for most purposes. The data were obtained from *Mathematica*'s curated databases.

paragraphs

A specialized plot for bivariate time series called the cave plot (Becker et al., 1994) is easily constructed in R as shown by Zhou and Braun (2010). When there are many multivariate time series, using `xyplot` may not feasible. In this case, `mvtsplot()` provided by Peng (2008) may be used. Many interesting examples including a stock market portfolio, daily time

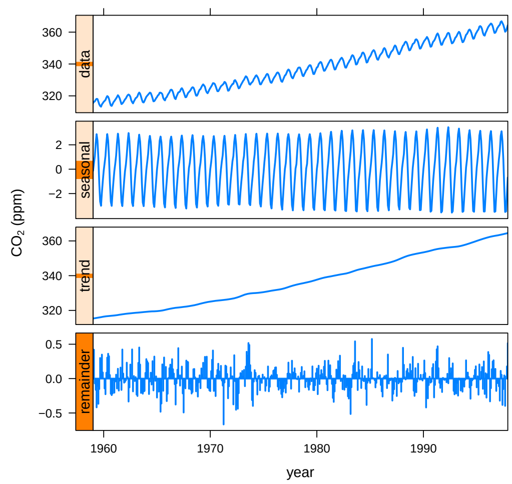


Figure 4: Atmospheric concentration of $CO_2$.

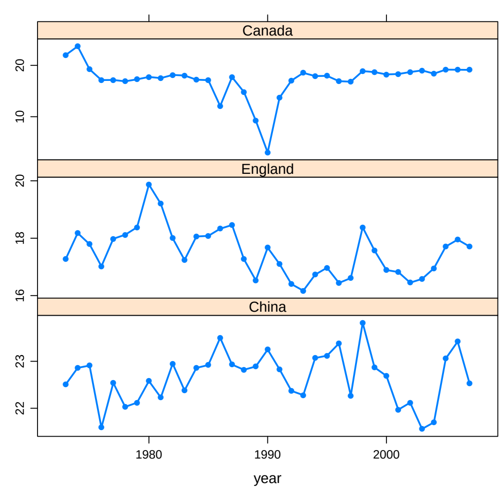


Figure 5: Average annual temperature Canada, England and China.

series of ozone pollution in 100 US counties and levels of sulfate in 98 US counties are discussed Peng (2008). For comparison, the previous annual temperature series is displayed using `mvtsplot()` in Figure 6. The right panel of the plot shows a boxplot for the values in each series. From this panel it is clear than China is generally much warmer than England and Canada and that England is often slightly cooler than Canada on an average annual basis. The bottom panel shows the average of the three series. The image shown shows the variation in the three series. The colors purple, grey and green correspond to low, medium and high values for each series. The darker the shading the larger the value. From image in Figure 6, it is seen that Canada has experienced relatively warmer years than England or China since about the year 2000. During 1989 to 1991 the average annual temperature in Canada was relatively low compared to England and China. There are many more possible option choices for constructing these plots (Peng, 2008). Financial time series are often

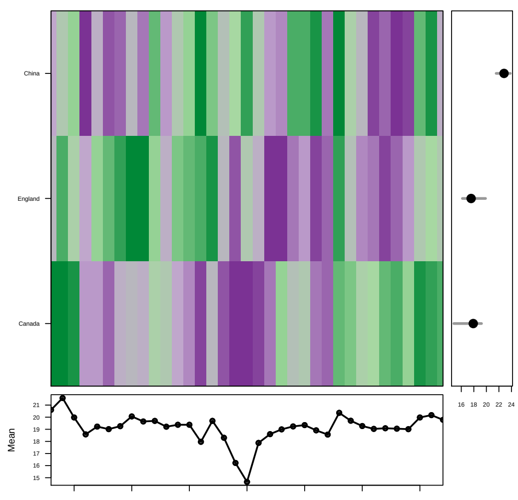


Figure 6: Average annual temperature in F°, 1973-2007.

observed on a daily basis but not including holidays and other days when the exchange is closed. Historical stock market data may be accessed using `get.hist.quote()` (Trapletti, 2011). Dealing with dates and times is often an important practical issue with financial time series. Grolemund and Wickham (2011) provide an new approach to this problem and review the other approaches that have been used in R. Irregularly observed time series can be plotted provided the dates are known using `timeSeries()` (Wuertz and Chalabi, 2011). The RMetrics packages **fImport** has functions for retrieving stock market data from various stock exchanges around the world. The function `yahooSeries()` is used to obtain the last 60 trading days of the close price of IBM stock and then `timeSeries()` is used to create an RMetrics time-series object containing the date information. This plot of IBM close price is shown in Figure 7. R provides excellent time

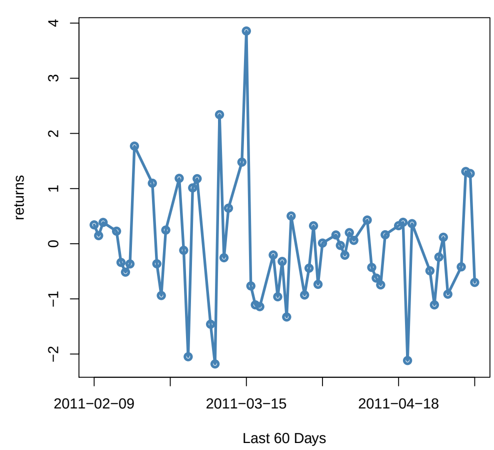


Figure 7: IBM, daily close price, returns in percent.

series graphic capabilities. Many of the standard functions R for autcorrelations, spectral analysis and wavelet decomposition to name a few are best understood from their associated graphical output. All figures in Cleveland (1993) may be reproduced using the R scripts available from http://www.stat.purdue.edu/~wsc/visualizing.html. The R package ggplot2 (Wickham, 2009) implements the novel graphical methods discussed by (Wilkinson, 1999). An interesting rendition of Millard's famous temporal-spacial graph of Napoleon's invasion of Russia is given using the **ggplot2**. The **rggobi** (Cook and Swayne, 2007) provides methods for dynamic data visualization including animated time series plots. The foundation of the R graphics system is described by in the book by Murrell (2005).

## 3. Base packages: stats and datasets

The **stats** and **datasets** packages are normally automatically loaded by default when R is started. These packages provide a comprehensive suite for functions for analyzing time series along with many interesting and well known time series datasets. The **stats** package provides the base functions for time series analysis. For discussion of some of the functions listed below see Cowpertwait and Metcalfe (2009). Some time series textbooks provide a brief introduction to R and its use for time series analysis (Cryer and Chan, 2008; Shumway and Stoffer, 2011; Venables and Ripley, 2002; Wuertz and core team members, 2010). Adler (2009) provides a comprehensive introduction to R and includes with a chapter on time series analysis. The **MASS** accompanies the textbook of  that also contains an introduction to spectral analysis and ARIMA models. The time series analysis functions that R provides are sufficient to provide a software supplement to most of the widely use textbooks for time series analysis

### *3.1.* stats

First we discuss the **stats** time series functions. In addition to many functions for manipulating time series such as filtering, differencing, inverse differencing, windowing, simulating, aggregating and forming multivariate series, there is a complete set of functions for auto/cross correlations analysis, seasonal decomposition using moving-average filters or loess, univariate and multivariate spectral analysis, univariate and multivariate autoregression, and univariate ARIMA model fitting with regression. Many of these functions implement state-of-the art algorithms. The `ar` function includes options, in both the univariate and multivariate cases, for Yule-Walker, least-squares or Burg estimates. Although `ar` can also do maximum likelihood estimator, the package **FitAR** (McLeod et al., 2011b; McLeod and Zhang, 2008b) provides a much faster and more reliable algorithm. The function `spectrum()`, also for both univariate and multivariate cases, implements the iterated Daniel smoother (Bloomfield, 2000). In the univariate case, the autoregressive spectral density estimator (Percival and Walden, 1993). The `arima()` function implements a Kalman filter algorithm that provides exact maximum likelihood estimation and an exact treatment for the missing-values (Ripley, 2002). This function is interfaced to C code to provide maximum computational efficiency. The `arima()` function has options for multiplicative seasonal ARIMA model

this is not
a sentence

fitting, subset models where some parameters are fixed at zero, and regression with ARIMA errors. The functions `tsdiag()` and `Box.test()` provide model diagnostic checks. For ARMA models, a new maximum likelihood algorithm (McLeod and Zhang, 2008a) written entirely in the R language is available in the **FitARMA** package (McLeod, 2010). A brief example of a medical intervention analysis carried out using `arima()` will now be discussed. In a medical time series of monthly average Creatinine clearances a step intervention analysis model with a multiplicative seasonal ARIMA$(0,1,1)$ $(1,0,0)_{12}$ error term was fit. The intervention effect was found to be significant at 1%. To illustrate this finding, Figure 8 compares the forecasts before and after the intervention date. The forecasts are from a model fit to the pre-intervention series. The plot visually confirms the decrease in Creatinine clearances after the intervention.

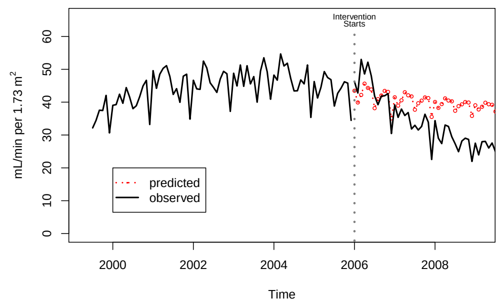


Figure 8: Creatinine clearance series.

Exponential smoothing methods are widely used for forecasting (Gelper et al., 2010) and are available in the **stats** (Meyer, 2002). Simple exponential smoothing defines the prediction for $z_{t+h}, h = 1, 2, \ldots$ as $\hat{z}_{t+1}$ where $\hat{z}_{t+1} = \lambda z_t + (1 - \lambda)\hat{z}_{t-1}$. Thus the forecast with this method is equivalent to that from an ARIMA(0,1,1). An extension double exponential smoothing forecasts $z_{t+h}, h = 1, 2, \ldots$ using the equation $\hat{z}_{t+h} = \hat{a}_t + h\hat{b}_t$, where $\hat{a}_t = \alpha z_t + (1 - \alpha)(\hat{a}_{t-1} + \hat{b}_{t-1}$, $\hat{b}_t = \beta(\hat{a}_t - \hat{a}_{t-1}) + (1 - \beta)\hat{b}_{t-1}$, $\alpha$ and $\beta$ are smoothing parameters. Double exponential smoothing is sometimes called Holt's linear trend method and it can be seen to produce forecasts equivalent to the ARIMA(0,2,2). The Winter's method for seasonal time series with period $p$, forecasts $z_{t+h}$, in the linear case can be written by $\hat{z}_{t+h} = \hat{a}_t + h\hat{b}_t + \hat{s}_t$, where $\hat{a}_t = \alpha(z_t - \hat{s}_{t-p}) + (1 - \alpha)(\hat{a}_{t-1} + \hat{b}_{t-1}$,

parentheses

$\hat{b}_t = \beta(\hat{a}_t - \hat{a}_{t-1}) + (1-\beta)\hat{b}_{t-1}$, $\hat{s}_t = \gamma(Y - \hat{a}_t) + (1-\gamma)\hat{s}_{t-p}$, $\alpha, \beta$ and $\gamma$ are smoothing parameters. In the multiplicative version, $\hat{z}_{t+h} = (\hat{a}_t + h\hat{b}_t)\hat{s}_t$. In the linear case, this is equivalent to the multiplicative seasonal ARIMA airline-model, All of the above modell may be fit with `HoltWinters()`. This function as `predict()` and `plot()` methods. Structural time series models (Harvey, 1989) are also implemented using Kalman filtering in the function `StructTS()`. Since the Kalman filter is used, Kalman smoothing is also available and it is implemented in the function `tsSmooth`. The basic structural model can be written is comprised of an observational equation,

$$z_t = \mu_t + s_t + e_t, \quad e_t \sim NID(0, \sigma_e^2)$$

and the state equations,

$$\mu_{t+1} = \mu_t + \mu_t + \xi_t, \quad \xi_t \sim \mathrm{NID}(0, \sigma_\zeta^2)$$

$$\nu_{t+1} = \nu_t + \zeta_t, \quad \zeta_t \sim \mathrm{NID}(0, \sigma_\zeta^2)$$

$$\gamma_{t+1} = -(\gamma_t + \ldots + \gamma_{t-s+2}) + \omega_t, \quad \omega_t \sim \mathrm{NID}(0, \sigma_\eta^2)$$

If $\sigma_\omega^2$ is set to zero, the seasonality is deterministic. The local linear trend model is obtained by omitting the term involving $\gamma_t$ in the observational equation and the last state equation may be dropped as well. Setting $\sigma_\zeta^2) = 0$ in the local linear trend model results in a model equivalent to the ARIMA(0,2,2) and setting $\sigma_\xi^2 = 0$ produces the local linear model that is equivalent to the ARMA(0,1,1). In Figure 9 the forecasts from the multiplicative Winter's method for the next 12 months are compared with forecasts from the multiplicative seasonal ARIMA$(0,1,1)\ (0,1,1)_{12}$ model. With this model logarithms of the original data were used and then the forecasts were back-transformed. In general, there are two types of backtransform that may be used for obtaining the forecasts in the original data domain (Granger and Newbold, 1976; Hopwood et al., 1984), described as naive or minimum-mean-square-error (MMSE). It is interesting that Figure 9 shows that the MMSE forecasts are shrunk relative to the others.

### *3.2.* Forecast

The package **Forecast** (Hyndman, 2010) provides further support for forecasting using ARIMA and a wide class of exponential smoothing models. These methods are described briefly by (Hyndman and Khandakar, 2008) and in more depth in the book (Hyndman et al., 2008). (Hyndman

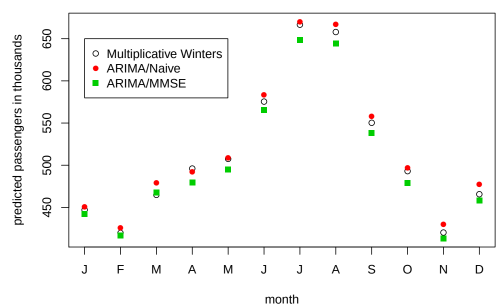


Figure 9: Comparisons of forecasts for 1961.

and Khandakar, 2008) provides a family of sixty different exponential smoothing models and provides a new state-space approach to evaluate the likelihood function that he defines for each model.

## 4. More Linear Time Series Analysis

### *4.1. Extensions for ARIMA and exponential smoothing*

The package **forecast** (Hyndman, 2010) provides further support for forecasting using ARIMA and a wide class of exponential smoothing models. These methods are descibed briefly by (Hyndman and Khandakar, 2008) and in more depth in the book (Hyndman et al., 2008). Hyndman and Khandakar (2008) provides a family of sixty different exponential smoothing models and provides a new state-space approach to evaluate the likelihood function that he defines for each model. General utility functions for that are useful for dealing with time series data such as number of days in each month, interpolation for missing values, a new seasonal plot, and others are briefly described in Table 4.1. Automatic ARIMA and related functions are summarized in Table 4.1. Table 4.1 summarizes functions for exponential smoothing models.

### *4.2. State space models and Kalman filtering*

Tusell (2011) provides an overview of Kalman filtering with R. In addition to `StructTS`, there are four other packages that support Kalman filtering and state-space modelling of time series. In general the state space model

| Function | Purpose |
|---|---|
| `accuracy()` | accuracy measures of forecast |
| `BoxCox, invBoxCox()` | Box-Cox transformation |
| `decompose()` | improved version of decompose() |
| `dm.test()` | Diebold-Mariano test compares the forecast accuracy |
| `forecast()` | generic function with various methods |
| `monthdays()` | number of days in seasonal series |
| `na.interp()` | interpolate missing values |
| `naive(), snaive()` | ARIMA(0,1,0) forecast and seasonal version |
| `seasadj()` | seasonally adjusted series |
| `seasonaldummy()` | create matrix of seasonal indicator variables |
| `seasonplot()` | season plot |

Table 1: General purpose utility functions.

| Function | Purpose |
|---|---|
| `arfima` | automatic ARFIMA |
| `Arima` | improved version of arima() |
| `arima.errors` | removes regression component |
| `auto.arima` | automatic ARIMA modelling |
| `ndiffs` | use unit root test to determine differencing |
| `tsdisplay()` | display with time series plot, acf, pacf, etc. |

Table 2: ARIMA functions.

| Function | Purpose |
|---|---|
| `croston` | Exponential forecasting for intermittent series |
| `ets` | Exponential smoothing state space model |
| `logLik.ets` | loglikelihood for ets object |
| `naive(), snaive()` | ARIMA(0,1,0) forecast and seasonal version |
| `rwf()` | random walk forecast with possible drifts |
| `ses(), holt(), hw()` | exponential forecasting methods |
| `simulate.ets()` | simulation method for ets object |
| `sindexf` | seasonal index, future periods |
| `splinef` | forecast using splines |
| `thetaf` | forecast using theta method |
| `tslm()` | lm()-like function using trend and seasonal |

Table 3: Exponential smoothing and other time series modelling functions.

(Harvey, 1989; Tusell, 2011) is comprised of two equations, the observation equation:

$$\boldsymbol{y}_t = \boldsymbol{d}_t + \boldsymbol{Z}_t \boldsymbol{\alpha}_t + \boldsymbol{\epsilon}_t \tag{1}$$

and state equation:

$$\boldsymbol{\alpha}_t = \boldsymbol{c}_t + \boldsymbol{T}_t \boldsymbol{\alpha}_{t-1} + \boldsymbol{R}_t \boldsymbol{\eta}_t, \tag{2}$$

where the white noises $\boldsymbol{\epsilon}_t$ and $\boldsymbol{\eta}_t$ are multivariate normal with mean vector zero and covariate matrices $\boldsymbol{Q}_t$ and $\boldsymbol{H}_t$ respectively and are uncorrelated. The Kalman filter algorithm can be used to recursively obtain

- predictions for $\boldsymbol{\alpha}_t$
- predictions for $\boldsymbol{y}_t$
- interpolation for $\boldsymbol{y}_t$

and in each case the estimated covariance matrix may also be obtained. Dropping the terms $\boldsymbol{d}_t$ and $\boldsymbol{c}_t$ and restricting all the matrices to be constant over time provides a class of state-space models that includes univariate and multivariate ARMA models (Brockwell and Davis, 1991; Durbin and Koopman, 2001). The built-in function `arima` also uses a Kalman filter algorithm (Ripley, 2002). The dse package Gilbert (2011) implements Kalman filtering for this time-invariant case and provides a general class of models that includes multivariate ARMA and ARMAX models. Harrison and West (1997) and Harvey (1989) provide a comprehensive account of Bayesian analysis dynamic linear models based on the Kalman filter and this theme is further developed in the book Petris et al. (2009) which also provides illustrative R scripts and code. The accompanying package **dlm** (Petris, 2010) provides functions for estimation and filtering as well as well-written vignette explaining how to use the software. The following example of fitting the random walk plus noise model,

$$
\begin{aligned}
y_t &= \theta_t + v_t, & v_t &\sim \mathcal{N}(0, V) \\
\theta_t &= \theta_{t-1} + w_t, & w_t &\sim \mathcal{N}(0, W)^{\cdot}
\end{aligned}
$$

to the Nile series and plotting the filtered series, Figure 10 and its 95% interval, is taken from the vignette by Petris (2010). Three other packages

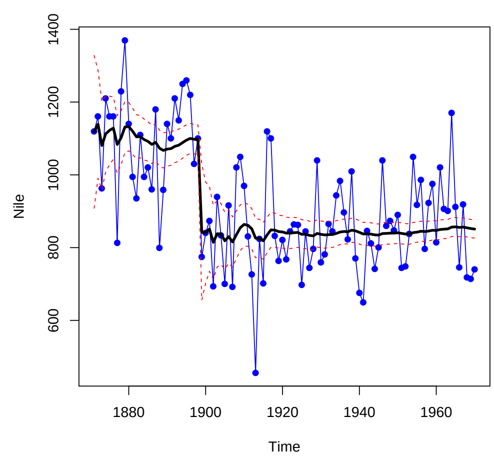


Figure 10: Nile river flows (solid line with circles), filter values after fitting random walk with noise (solid thick line) and 95% confidence interval (dashed lines)

for Kalman filtering (Dethlefsen et al., 2009; Luethi et al., 2010; Helske, 2011) are also reviewed by Tusell (2011).

*4.3. An approach to linear time series analysis using Durbin-Levinsion recursions*

The Durbin-Levinson recursions Box et al. (2008) provide a simple and direct approach to the computation of the likelihood, computation of exact forecasts and their covariance matrix, and simulation that is implemented in **ltsa** (McLeod et al., 2007, 2011a). In Section 4.4 this approach is implemented for the fractional Gaussian noise (FGN) and a comprehensive model building R package is provided for this purpose using the functions in **ltsa**. Three methods of simulating a time series given its autocovariance

| Function | Purpose |
|---|---|
| `DHSimulate` | Simulate using Davies-Harte method |
| `DLLoglikelihood` | Exact concentrated log-likelihood |
| `DLResiduals` | Standardized prediction residuals |
| `DLSimulate` | Simulate using DL recursion |
| `SimGLP` | Simulate general linear process |
| `TrenchInverse` | Toeplitz matrix inverse |
| `ToeplitzInverseUpdate` | Updates the inverse |
| `TrenchMean` | Exact MLE for mean |
| `TrenchForecast` | Exact forecast and variance |

Table 4: The principal functions in **ltsa**.

function are available: `DHSimulate()`, `DLSimulate()`, and `SimGLP()`. `DHSimulate()` implements the fast Fourier algorithm (FFT) of Davies and Harte (1987). But this algorithm is not applicable for all stationary series (Craigmile, 2003) so `DHSimulate()`, based on the Durbin-Levinson recursion, is also provided. The algorithm `SimGLP` is provided for simulating a time series with non-Gaussian innovations based on the equation,

$$z_t = \mu + \sum_{i=1}^{Q} \psi_i a_{t-i}. \tag{3}$$

The sum involved in Equation (3) is efficiently evaluated using the R function `convolve()` using the FFT method. The built-in function `arima.sim()` may also be used in the case of ARIMA models. The functions `TrenchInverse()` and `TrenchInverseUpdate()` are useful in some applications involving Toeplitz covariance matrices. `TrenchForecast` provides exact forecasts and their covariance matrix. The following illustration often useful in time series lectures when forecasting is discussed. In this example we fit an AR(9) to the annual sunspot numbers, 1700-1988, `sunspot.year`. As is usual, we regard these parameters as known for forecasting. Letting $z_m(\ell)$ denote the optimal minimum mean square error forecast at origin time $t = m$ and lead time $\ell$, we compare the forecasts of $z_{m+1}, \ldots, z_n$ using the one-step ahead predictor $z_{m+\ell-1}(1)$, with the fixed origin prediction $z_m\ell$, where $\ell = 1, \ldots, L$ and $L = n - m + 1$. Figure 11 compares forecasts and we see many interesting features. The fixed origin forecasts are less accurate as expected from theory. As well the fixed origin forecasts show systematic departures whereas the one-step do not. As

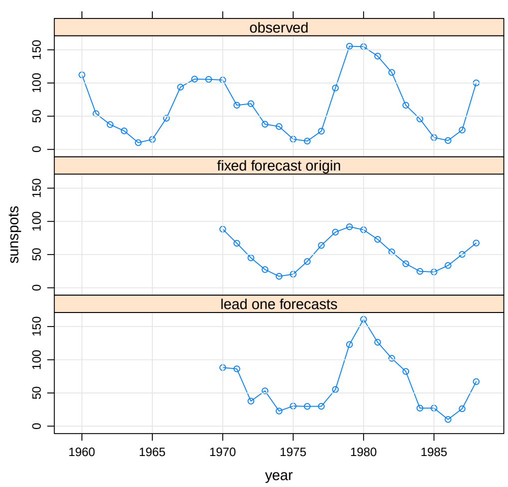


Figure 11: Comparing forecasts from a fixed origin, 1969, with lead-one forecasts starting in 1969 for `sunspot.year`.

shown by this example, `TrenchForecast` provides a more flexible approach to forecasting than provided by `predict`.

### *4.4. Long memory time series analysis*

Let $z_t$, $t = 1, 2, \ldots$ be stationary with mean zero and autocovariance function, $\gamma_z(k) = \mathrm{cov}(z_t, z_{t-k})$. many long memory processes such as the FGN (fractional Gaussian Noise) and FARMA (fractional ARMA) may be characterized by the property that $k^{\alpha}\gamma_Z(k) \to c_{\alpha,\gamma}$ as $k \to \infty$, for some $\alpha \in (0,1)$ and $c_{\alpha,\gamma} > 0$. Equivalently,

$$\gamma_Z(k) \sim c_{\alpha,\gamma}\, k^{-\alpha}.$$

The FARMA and FGN models are reviewed by Hipel and McLeod (1994); Beran (1994); Brockwell and Davis (1991). FGN can simply be described as a stationary Gaussian time series with covariance function, $\rho_k = (|k+1|^{2H} - 2|k|^{2H} + |k-1|^{2H})/2,\ 0 < H < 1$. The FARMA model essentially generalizes the ARIMA model to a family of stationary models with fractional difference parameter $d$, $d \in (-0.5, 0.5)$. The long-memory parameters $H$ and $d$ may be expressed in terms of $\alpha$, $H \simeq 1 - \alpha/2,\ H \in (0,1),\ H \neq 1/2$ and $d \simeq 1/2 - \alpha/2,\ d \in (-1/2, 1/2),\ d \neq 0$ (McLeod, 1998). Gaussian white noise corresponds to $H = 1/2$ and in the case of FARMA, $d = 0$ assuming no AR or MA components. (Haslett and Raftery, 1989) developed an algorithm for maximum likelihood estimation of FARMA models and applied these models to the analysis of wind speed time series with $n > 30000$. This algorithm is available in R in the package `fracdiff` (Fraley et al., 2009). The generalization of the FARMA model to allow more general values of $d$ is usually denoted by ARFIMA. A frequently cited example of a long-memory time is the minimum annual flows of the Nile over the period 622-1284, $n = 663$ (Percival and Walden, 2000, §9.8). The package `longmemo` (Beran et al., 2009) has this data as well as other time series examples. **FGN** provides exact MLE for the parameter $H$ as well as a parametric bootstrap and minimum square error forecasts. For the Nile data, $\hat{H} = 0.831$. The time series plots in Figure 12 show the actual Nile series along with three bootstraps. As a further illustration of capabilities of R, a simulation experiment was done to compare the estimation of the $H$-parameter in fractional Gaussian noise using exact MLE function `FitFGN()` in **FGN** and the GLM method `FEXPest()` in the package **longmemo**. The function `SimulateFGN` in **FGN** was used to simulate 100 sequences of length $n = 200$ for $H = 0.3, 0.5, 0.7$. Each sequence was fit by the MLE and GLM method and the absolute error of the difference between the estimate and the true parameter was obtained, that is, $\mathrm{Err_{MLE}} = |\hat{\mathrm{H}}_{\mathrm{MLE}} - \mathrm{H}|$ and $\mathrm{Err_{GLM}} = |\hat{\mathrm{H}}_{\mathrm{GLM}} - \mathrm{H}|$. Figure 13, shows the a notched boxplot of the 100 differences $\mathrm{Err_{(GLM)}} - \mathrm{Err_{(MLE)}}$. Figure 13 shows that the MLE is more accurate. These computations take less than 5 seconds using sequential evaluation on a current computer. The ARFIMA model extends the stationary fractional models to the one stationary case

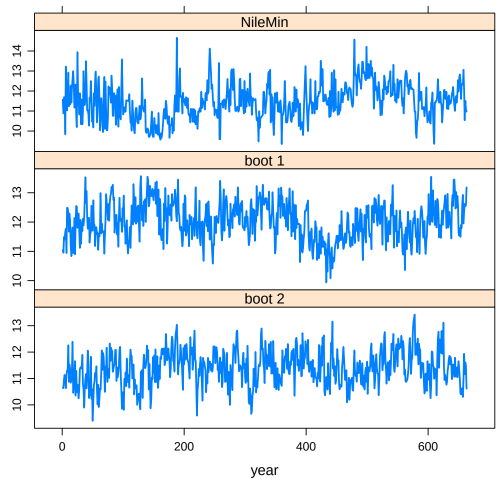


Figure 12: Comparing actual Nile minima series with two bootstrap versions.

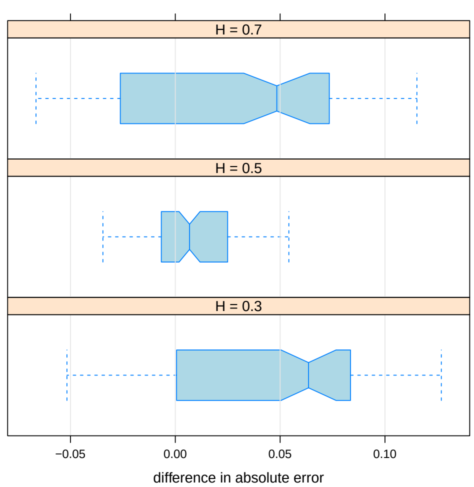


Figure 13: Comparing MLE estimator and GLM estimator for the parameter H in fractional Gaussian noise.

(Diebold and Rudebusch, 1989; Baillie, 1996). The simplest approach is to choose the differencing parameter and then fit the FARMA model to the differenced time series.

### 4.5. Subset autoregression

This package provides a more efficient and reliable exact MLE for AR(p) than is available with the built-in function `ar()`. Two types of subset autoregressions may also be fit. The usual subset autoregression may be written, $\phi(B)(z_t - \mu) = a_t$, where $\phi(B) = 1 - \phi_{i_1} B - \ldots - \phi_{i_m} B^{i_m}$, where $i_1, \ldots, i_m$ are the subset of lags. For this model OLS is used to estimate the parameters. The other subset family is parameterized using the partial autocorrelations as parameters. Efficient model selection, estimation and diagnostic checking algorithms are discussed by McLeod and Zhang (2006, 2008b) and implemented in the package `FitAR` (McLeod et al., 2011b). Any stationary time series can be approximated by a high order autoregression that may be selected using one of several information criteria. Using this approximation, `FitAR`, provides functions for automatic bootstrapping, spectral density estimation, and Box-Cox analysis for any time series. The optimal Box-Cox transformation for the `lynx` is obtained simply from the command `R > BoxCox(lynx)` and plot showing the result is shown in Figure 14. The main functions of interest are listed in the Appendix.

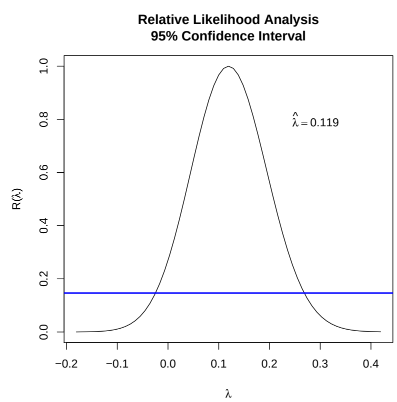


Figure 14: Box-Cox analysis of lynx time series.

*4.6. Periodic autoregression*

Let $z_t, \quad t = 1, \ldots, n$ be $n$ consecutive observations of a seasonal time series with seasonal period $s$. For simplicity of notation, assume that $n/s = N$ is an integer, so $N$ full years of data available. The time index parameter, $t$, may be written $t = t(r, m) = (r - 1)s + m$, where $r = 1, \ldots, N$ and $m = 1, \ldots, s$. In the case of monthly data $s = 12$ and $r$ and $m$ denote the year and month. If expected monthly mean $\mu_m = \mathrm{E}\{z_{t(r,m)}\}$ and the covariance function, $\gamma_{\ell,m} = \mathrm{cov}(z_{t(r,m)}, z_{t(r,m)-\ell})$ depend only on $\ell$ and $m$, $z_t$ is said to be periodically correlated and is periodic stationary. The periodic AR model of order $(p_1, \ldots, p_s)$ may be written,

$$z_{t(r,m)} = \mu_m + \sum_{i=1}^{p_m} \phi_{i,m}(z_{t(r,m)-i} - \mu_{m-i}) + a_{t(r,m)}, \tag{1.3}$$

where $a_{t(r,m)} \sim \mathrm{NID}(0, \sigma_m^2)$, where $m$ obeys modular arithmetic. This model is model originated in modelling monthly streamflow and is further discussed with examples (Hipel and McLeod, 1994; McLeod, 1994). The package `pear` (McLeod and Balcilar, 2011) implements functions for model identification, estimation and diagnostic checking for periodic AR models. We conclude with a brief mention of some recent work on periodically correlated time series models which we hope to see implemented in R. Tesfaye et al. (2011) develop a parsimonious and efficient procedure for dealing with periodically correlated daily ARMA series and provide applications to geophysical series. Ursu and Duchesne (2009) extend modelling procedures the vector PAR model and provide an application to macro economic series. Aknouche and Bibi (2009) show that quasi-MLE provide consistent asymptotically normal estimates in an periodic GARCH model under mild regularity conditions.

## 5. Time series regression

An overview of some selected time series regression topics is given in this section. For additional time series regression topics and discussion of related R software is available in several books (Cowpertwait and Metcalfe, 2009; Cryer and Chan, 2008; Kleiber and Zeileis, 2008; Shumway and Stoffer, 2011).

*5.1. Cigarette consumption data*

Most of the regression methods discussed in this section will be illustrated with data from an empirical demand analysis for cigarettes in Canada

(Thompson and McLeod, 1976). The variables of interest, consumption of cigarettes per capita, $Q_t$, real disposable income per capita, $Y_t$, and the real price of cigarettes, $P_t$, for $t = 1, \ldots, 23$ corresponding to the years 1953-1975 were all logarithmically transformed, input, and converted to a dataframe `cig`. For some modeling purposes, it is more convenient to use a `ts` object,

```
R >cig.ts <- ts(as.matrix.data.frame(cig), start = 1953,
+     freq = 1)
```

The time series are shown in Figure 15.

```
R >plot(cig.ts, xlab = "year", main = "", type = "o")
```

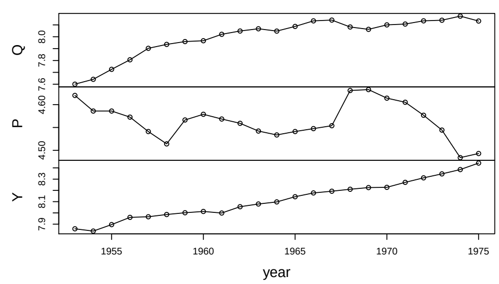


Figure 15: Canadian cigarette data. Consumption/adult(Q), real price(P), income/adult(Y)

*5.2. Durbin-Watson test*

The exact p-value for the Durbin-Watson diagnostic test for lack of autocorrelation in a linear regression with exogenous inputs and Gaussian white noise errors is available with the function `dwtest()` in the **lmtest** package (Hothorn et al., 2010). The diagnostic check statistic may be written

$$d = \frac{\sum_{t=2}^{n} (\hat{e}_t - \hat{e}_{t-1})^2}{\sum_{t=1}^{n} \hat{e}_t^2}, \tag{4}$$

where $\hat{e}_t, t = 1, \ldots, n$ are the OLS residuals. Under the null hypothesis, $d$ should be close to 2 and small values of $d$ indicate positive autorrelation. Many econometric textbooks provide tables for the critical values of $d$. But if you run this beta_2=1 and everything else should be 0 - what is Y

in small samples these tables may be inadequate since there is a fairly large interval of values for $d$ for which the test is inconclusive. This does not happen when the exact p-value is computed. Additionally, current statistical practice favours reporting p-values in diagnostic checks (Moore, 2007). The Durbin-Watson test is very useful in time series regression for model selection. When residual autocorrelation is detected, sometimes simply taking first or second differences is all that is needed to remove the effect of autocorrelation. In the next example we find that taking second differences provides an adequate model.

too many Qs
what is y
no mention of
cautions/drawbacks
of lm() for time series
lagged variables,
and na.action

First we fit the empirical demand equation, $Q_t = \beta_0 + \beta_1 P_t + \beta_2 Q_t + e_t$ using OLS with the `lm()` function. Some of the output is shown below.

```
               Estimate Std. Error    t value     Pr(>|t|)
(Intercept)  3.328610  2.5745756  1.2928771 2.107900e-01
P           -0.402811  0.4762785 -0.8457468 4.076991e-01
Y            0.802143  0.1118094  7.1741970 6.011946e-07
```

This output suggests $P_t$ is not significant but $Y_t$ appears to be highly significant. But the Durbin-Watson test rejects the null hypothesis of no autocorrelation. After differencing the null hypothesis is still rejected. Fitting the model, $\nabla^2 Q_t = \beta_0 + \nabla^2\beta_1 P_t + \nabla^2\beta_2 Q_t + e_t$, $\hat{\beta}_1 = 0.557$ with a 95% margin of error, 0.464. The computations and results are shown below,

```
R >cig2.lm <- lm(Q ~ P + Y, data = diff(cig.ts, differences = 2))
R >summary(cig2.lm)$coefficients
```

```
                Estimate  Std. Error    t value   Pr(>|t|)
(Intercept) -0.003118939 0.008232764 -0.3788447 0.70923480
P           -0.557623890 0.236867207 -2.3541625 0.03012373
Y            0.094773991 0.278979070  0.3397172 0.73800132
```

The intercept term, corresponds to a quadratic trend, is not signficant and can be dropped. Income, $Y_t$ is also not significant. The evidence for lag-one autocorrelation is not strong,

```
R >dwtest(cig2.lm, alternative = "two.sided")

        Durbin-Watson test

data:  cig2.lm
DW = 2.6941, p-value = 0.08025
alternative hypothesis: true autocorelation is not 0
```

if x_t = t then you will get trend, so I don't understand the ref to forecast

it also reads like you can only fit 2 indep. variables.

There is also no evidence of non-normality using the Jarque-Bera test. We use the function `jarque.bera.test()` in the **tseries** package (Trapletti, 2011).

```
R >jarque.bera.test(resid(cig2.lm))

        Jarque Bera Test

data:  resid(cig2.lm)
X-squared = 1.1992, df = 2, p-value = 0.549
```

Kleiber and Zeileis (2008, §7.1) conduct an illustrative simulation experiment using R that compares the power of the Durbin-Watson test with the Breusch-Godfrey test.

*5.3. Regression with autocorrelated error*

The built-in function `arima` can fit the regression model $y_t = \beta_0 + \beta_1 x_{1,t} + \beta_2 x_{2,t} + e_t$, where $e_t \sim \mathrm{ARIMA(p, d, q)}$. The `Arima` function in the package **forecast** (Hyndman, 2010) can include a linear trend. We illustrate by fitting an alternative to the regression just fit above for the Canadian cigarette data.

```
R >with(cig, arima(Q, order = c(1, 1, 1), xreg = cbind(P,
+     Y)))

Call:
arima(x = Q, order = c(1, 1, 1), xreg = cbind(P, Y))

Coefficients:
         ar1      ma1        P       Y
      0.9332  -0.6084  -0.6718  0.2988
s.e.  0.1010   0.2007   0.2037  0.2377

sigma^2 estimated as 0.0008075:  log likelihood = 46.71,  aic = -83.41
```

This model agrees well with the linear regression using second differencing.

another goof

### 5.4. Regression with lagged variables

Linear regression models with lagged dependent and/or independent variables are easily fit using the **dynlim** package (Zeileis, 2010). In the case of the empirical demand for cigarettes, it is natural to consider the possible effect lagged price. $\nabla^2 Q_t = \beta_1 \nabla^2 P_t + \beta_{1,2} \nabla^2 P_{t-1} + \beta_2 \nabla^2 Q_t + e_t$,

```
R >summary(dynlm(Q ~ -1 + P + L(P) + Y, data = diff(cig.ts,
+     differences = 2)))$coefficients

       Estimate Std. Error    t value   Pr(>|t|)
P    -0.6421079  0.2308323 -2.7817077 0.01278799
L(P) -0.1992065  0.2418089 -0.8238177 0.42145104
Y    -0.2102738  0.2993858 -0.7023507 0.49196623
```

We see that lagged price is not significant.

### 5.5. Structural Change

Brown et al. (1975) introduced recursive residuals and related methods for examining graphically the stability of regression over time. These methods and recent developments in testing and visualizing structural change in time series regression are discussed in the book by Kleiber and Zeileis (2008, §6.4) and implemented in the package **strucchange** (Zeileis et al., 2010, 2002). We use a CUMSUM plot of the recursive residuals to check the regression using second differences for stability. No instability is detected with this analysis.

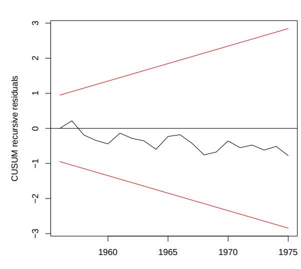


Figure 16: Cusum test of residuals in cigarette demand regression.

*5.6. Generalized linear models*

Kedem and Fokianos (2002) provides a mathematical treatment of the use of generalized linear models (GLM) for modelling stationary binary, categorical and count time series. GLM models can account for autocorrelation by using lagged values of the dependent variable in the systematic component. Under regularity conditions Kedem and Fokianos (2002, §1.4), inferences, based on large sample theory, for GLM time series models can be made using standard software that is used for fitting regular GLM models. In R, the function `glm()` may be used and it is easy to verify estimates of the precision using the `boot()` function. These GLM-based time series models are extensively used with longitudinal time series (Li, 1994). As an illustration, we consider the late night fatality data discussed in Vingilis et al. (2005). The purpose of this analysis was to investigate the effect of the extension of bar closing hours to 2:00 AM that was implemented May 1, 1996. This type of intervention analysis (Box and Tiao, 1975) is known as an interrupted time series design in the social sciences (Shadish et al., 2001). The total fatalities per month for the period starting January 1992 and through to December 1999, corresponding to a time series of length $n = 84$, are shown in Figure 17. The output from the

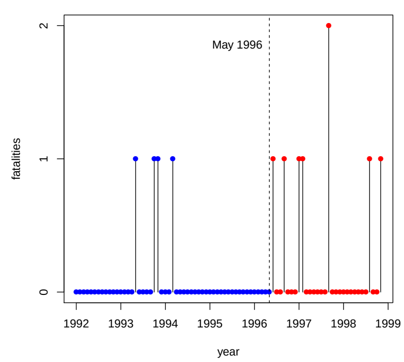


Figure 17: Late night car fatalities in Ontario. Bar closing hours were extended May 1996.

`glm()` function using `y` as the dependent variable, `y1` as the lagged dependent variable, and `x` as the step intervention defined as as `0` before May 1, 1996 and `1` after.

```
R >summary(ans)$coefficients
```

```
              Estimate Std. Error     z value     Pr(>|z|)
(Intercept) -2.53923499  0.5040873 -5.03729193 4.721644e-07
x2           1.16691417  0.6172375  1.89054329 5.868534e-02
y1          -0.06616152  0.6937560 -0.09536712 9.240232e-01
```

The resulting GLM model may be summarized as follows. The total fatalities per month, $z_t$, are Poisson distributed with mean $\mu_t$, where $\hat{\mu}_t = \exp\{\hat{\beta}_0 + \hat{\beta}_1 x_t + \hat{\beta}_2 y_{t-1}\}$, $\hat{\beta}_0 \doteq -2.54$, $\hat{\beta}_1 \doteq 1.17$, and $\hat{\beta}_2 \doteq -0.07$. We verified the standard deviation estimates of the parameters by using a non-parametric bootstrap with 1000 bootstrap samples. This computation takes less than 10 seconds on most current PC's.

```
(intercept)          x2          y1
  0.4921328   0.6614443   0.7485225
```

There is no evidence of lagged dependence but the intervention effect, $\beta_2$ is significant with $p < 0.10$. Hidden Markov models provide another time series generalization of Poisson and binomial GLM models (Zucchini and MacDonald, 2009).

## 6. Nonlinear time series models

Volatility models including the GARCH family of models are one of the newest types on nonlinear time series models. Nonlinear regression models can sometimes be applied to time series. GLM models provide an extension of linear models that is useful for modelling logistic and count time series (Kedem and Fokianos, 2002) Ritz and Streibig (2008) provides an overview nonlinear regression models using R. Loess regression in R provides a flexible nonparametric regression approach to handling up to three inputs but by using generalized additive models, many more inputs could be accommodated Wood (2006). Two packages Milborrow (2011); Hastie and Tibshirani (2011) implement the multiadaptive regression splines, MARS, method developed by Friedman (1991). It was was found by Lewis and Stevens (1991) that MARS regression produced better out-of-sample forecasts for the the annual sunspot series than competing nonlinear models. In the remainder of the section we discuss tests for nonlinearity and two popular approaches to modelling and forecasting nonlinear time series, threshold autoregression and neural net.

### 6.1. *Tests for nonlinear time series*

One approach is to fit a suitable ARIMA or other linear time series model and then apply the usual Ljung-Box portmanteau test to the squares of the residuals. McLeod and Li (1983) suggested this as a general test for nonlinearity. The built-in function `Box.test()` provides a convenient function for performing this test. Two tests (Teraesvirta et al., 1993; Lee et al., 1993) for neglected nonlinearity that are based on neural nets are implemented in **tseries** (Trapletti, 2011) as functions `terasvirta.test()` and `white.test()`. The Keenan test for nonlinearity Keenan (1985) is available in **TSA** (Chan, 2011) and is reviewed in the textbook by Cryer and Chan (2008).

### 6.2. *Threshold models*

Threshold autoregression (TAR) provides a general flexible family for nonlinear time series modelling that has proved useful in many applications. This approach is well suited to time series with stochastic cyclic effects such as exhibited in the annual sunspots or lynx time series. The model equation for a two-regime TAR model may be written,

$$\begin{aligned} y_t = {} & \phi_{1,0} + \phi_{1,1} y_{t-1} + \ldots + \phi_{1,p} y_{t-p} \\ & + I(y_{t-d} > r)\{\phi_{2,0} + \phi_{2,1} y_{t-1} + \ldots + \phi_{2,p} y_{t-p}\} + \sigma a_t \end{aligned} \tag{5}$$

where $I(y_{t-d} > r)$ indicates if $y_{t-d} > r$ the result is 1 and otherwise it is 0. The parameter $d$ is the delay parameter and $r$ is the threshold. There are separate autoregression parameters for each regime. This model may be estimated by least squares or more generally using conditional maximum likelihood. The predator time series in Figure 18 is described in the book by Cryer and Chan (2008) and is available in the companion package **TSA** (Chan, 2011). The package **TSA** provides the function `tar()` for fitting two regime TAR models and methods functions `predict()` and `tsdiag()` as well as functions `tar.skelton()` and `tar.sim()`. TAR and related models are also discussed by Tsay (2010) and some R scripts are provided as well the companion package **FinTS** (Graves, 2011) that includes data sets from the book. Figure 19 shows monthly U.S. unemployment. Tsay (2010, Example 4.2) fits the two regime TAR model,

$$\begin{aligned} y_t &= 0.083 y_{t-2} + 0.158 y_{t-3} + 0.0118 y_{t-4} - 0.180 y_{t-12} + a_{1,t} \quad \text{if } \mathrm{y}_{t-1} \le 0.01, \\ &= 0.421 y_{t-2} + 0.239 y_{t-3} - 0.127 y_{t-12} + a_{2,t} \quad \text{if } \mathrm{y}_{t-1} > 0.01, \end{aligned}$$

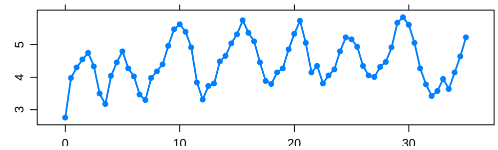


Figure 18: U.S. civilian unemployment rate, seasonally adjusted, January 1948 to March 2004

where $y_t$ is the differenced unemployment series. The estimated standard deviations of $a_{2,t}$ and $a_{2,t}$ were 0.180 and 0.217. Tsay (2010) remarks that the TAR provides more insight into the time-varying dynamics of the unemployment rate than the ARIMA.

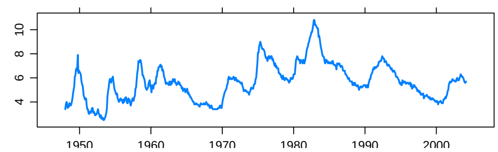


Figure 19: U.S. civilian unemployment rate, seasonally adjusted, January 1948 to March 2004

*6.3. Neural Nets*

Feed-forward neural networks provide another nonlinear generalization of the autogression model that has been demonstrated to work well in suitable applications (Faraway and Chatfield, 1998; Kajitani et al., 2005). Modelling and forecasting is easily done using **nnet** (Ripley, 2011). A feed-forward neural net that generalizes linear autoregressive model of order $p$ may be written,

$$y_t = f_o\left(a + \sum_{i=1}^{p} \Omega_i x_i + \sum_{j=1}^{H} w_j f\left(\alpha_j + \sum_{i=1}^{p} \omega_{i,j} x_{t-i}\right)\right) \tag{6}$$

where $\hat{y}_t$ is the predicted time series at time $t$ and $y_{t-1}, \ldots, y_{t-p}$ are the lagged inputs, $f_o$ is the activation function for the output node, $f$ is the activation function for each hidden node, $\omega_{i,j}$ are the $p$ weights along the connection for the $j$-th hidden node, $j = 1, \ldots, H$; $\Omega_i$ is the weight in the skip-layer connection connecting the input directly with the output $a$ is the bias connection. There are $m(1 + H(p + 2))$ unknown parameters that must be estimated. The hyperparameter $H$, the number of hidden nodes, is determined by a type of cross-validation and is discussed by Faraway and Chatfield (1998); Kajitani et al. (2005) in the time series context. The activation functions $f$ and $f_o$ are often chosen to be logistic, $\ell(x) = 1/(1 + e^{-x})$. A schematic illustration for $p = 2$ and $H = 2$ is shown in Figure 20. Feed-forward neural nets may be generalized for multivariate time series. Hastie et al. (2009) pointed out that the feed-forward neural net

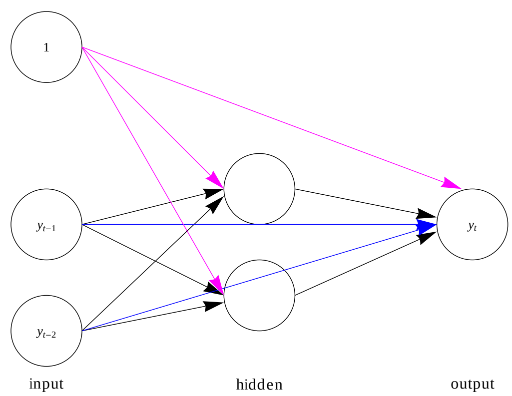


Figure 20: A nonlinear version of the AR(2) using the feedforward neural net.

defined in eqn. (6) is mathematically equivalent to the projection pursuit regression model. The net defined in eqn. (6) as well as the one illustrated in Figure 20 has just one hidden layer with $p$ and $p = 2$ nodes respectively. These nets may be generalized to accommodate more than one hidden layer and such nets provide additional flexibility. Ripley (1996) shows that asymptotically for a suitable number of hidden nodes, $H$, and a large enough training sample, the feed-forward neural net with one hidden layer can approximate any continuous mapping between the inputs and outputs.

## 7. Unit root tests

Financial and economic time series such as macro/micro series, stock prices, interest rates and many more, often exhibit nonstationary wandering behavior which is easily corrected by differencing. Such series are sometimes called homogeneous nonstationary or difference-stationary. Pretesting for a unit root is useful in ARIMA modelling and in cointegration modelling. Since actual time series may also exhibit other departures from the stationary Gaussian ARMA, many other types of unit root tests have been developed that are appropriate under various other assumptions (Said and Dickey, 1984; Phillips and Perron, 1988; Elliott et al., 1996; Kwiatkowski et al., 1992). These tests are all implemented in the R packages **fUnitRoots** (Wuertz et al., 2009b), **tseries** (Trapletti, 2011), and **urca** (Pfaff, 2010a).

### *7.1. Overview of the* **urca** *package*

The **urca** (Pfaff, 2010a) package offers a comprehensive and unified approach to unit root testing as discussed in the book Pfaff (2006). A flowchart for using the **urca** package to test for unit roots is given by Pfaff (2006, Chapter 5). Three regressions with autocorrelated AR(p) errors are considered for the unit root problem,

$$\Delta Z_t = \beta_0 + \beta_1 t + \gamma Z_{t-1} + \sum_{i=1}^{p-1} \delta_i \Delta Z_{t-i} + e_t \tag{7}$$

$$\Delta Z_t = \beta_0 + \gamma Z_{t-1} + \sum_{i=1}^{p-1} \delta_i \Delta Z_{t-i} + e_t, \tag{8}$$

$$\Delta Z_t = \gamma Z_{t-1} + \sum_{i=1}^{p-1} \delta_i \Delta Z_{t-i} + e_t, \tag{9}$$

corresponding respectively to unit root with drift constant plus deterministic trend, random walk with drift, and pure random walk. The test for unit root corresponds to an upper-tail test of $\mathcal{H}_0 : \gamma = 0$ and the parameters $\beta_0$ and $\beta_1$ correspond to the drift constant and deterministic time trend respectively. When $p = 1$, the test reduces to the standard Dickey-Fuller tests. The order of the autoregression is estimated using the AIC or BIC. For all three models the test statistic for testing $\mathcal{H}_0 : \gamma = 0$ is

$$\tau_i = \frac{\hat{\phi} - 1}{\mathrm{SE}(\hat{\phi})}, \quad i = 1, 2, 3,$$

where $i$ denotes the model (7), (8), or (9) respectively. The distribution of $\tau_i$ has been obtained by Monte-Carlo simulation or by response surface regression methods (MacKinnon, 1996). If $\tau_3$ is insignificant, so that $\mathcal{H}_0 : \gamma = 0$ is not rejected, the non standard $F-$statistics $\Phi_3$ and $\Phi_2$ are evaluated using the extra-sum-of-squares principle to test the null hypotheses $\mathcal{H}_0 : (\beta_0, \beta_1, \gamma) = (\beta_0, 0, 0)$ and $\mathcal{H}_0 : (\beta_0, \beta_1, \gamma) = (0, 0, 0)$ respectively. That is, to test whether the deterministic time trend term is needed in the regression model provided that the model has a drift term in the first hypothesis and has no drift term in the second one. If $\tau_2$ is insignificant, so that $\mathcal{H}_0 : \gamma = 0$ is not rejected, the non standard $F-$statistic $\Phi_1$ is evaluated using the extra-sum-of-squares principle to test the hypotheses $\mathcal{H}_0 : (\beta_0, \gamma) = (0, 0)$. That is, to test whether the regression model has a drift term. If $\mathcal{H}_0 : \gamma = 0$ is not rejected, we conclude that the series has a unit root. These steps may be repeated after differencing the series to test if further differencing is needed.

*7.1.1. Illustrative example*

As an example, consider the U.S. real GNP from 1909 to 1970 available as `nporg` in the **urca** package. Using the `lags = 4` in the argument to `ur.df`, the BIC selects $p = 1$. One advantage of using the `ur.df` function is that the regression and all test statistics may be displayed using the `summary` method for `ur.df`. When Sweave (Leisch, 2002) is used, Table 7.1.1 may be obtained directly from the output produced in R. Figure 21, shows the graphical model diagnostics. As previously noted in the introduction, all figures presented in the article may be generated using the R code supplement. In Step 2, the $\tau_3$ statistic for the null hypothesis $\gamma = 0$ is

| | Estimate | Std. Error | t value | Pr(>\|t\|) |
|---|---|---|---|---|
| (Intercept) | -0.331 | 4.025 | -0.082 | 0.935 |
| z.lag.1 | -0.043 | 0.033 | -1.308 | 0.196 |
| tt | 0.617 | 0.317 | 1.944 | 0.057 |
| z.diff.lag | 0.390 | 0.132 | 2.962 | 0.004 |

Table 5: Regression with constant and trend for the U.S. real GNP data starting at 1909 until 1970.

$-1.31$ and its corresponding critical values at levels $1\%, 5\%$, and $10\%$ with 62 observations are given in Table 7.1.1 as $-4.04, -3.45$, and $-3.15$ respectively. At these levels we can't reject the null of a unit root. Instead of comparing the test statistic value with the critical ones, one can use the MacKinnon's p-value determined from response surface regression methodology (MacKinnon, 1996). The function `punitroot()` is available in **urca**. In the present example, the p-value is $0.88$ and it corresponds to the $\tau_3$ statistic value confirming that the unit root hypothesis can not be rejected. The $F-$statistic $\Phi_3$ is used to test whether the deterministic time

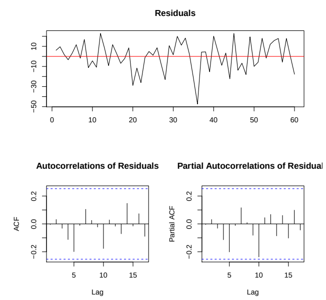


Figure 21: Residual diagnostic of U.S. real GNP data from 1909 to 1970.

| | 1pct | 5pct | 10pct |
|---|---|---|---|
| tau3 | -4.04 | -3.45 | -3.15 |
| phi2 | 6.50 | 4.88 | 4.16 |
| phi3 | 8.73 | 6.49 | 5.47 |

Table 6: Dickey-Fuller critical values for test statistics for drift and trend case.

trend term is needed in the regression model provided that the model has a drift term. The test statistic has a value of 2.68. From Table 7.1.1, the critical values of $\Phi_3$ at levels $1\%, 5\%$, and $10\%$ with 62 observations are $8.73, 6.49$, and $5.47$. We conclude that the null hypothesis is not rejected and a trend term is not needed. Thus we proceed to the next step and estimate the regression parameters in eqn. (8) with a drift term.
The $\tau_2$ statistic for the null hypothesis $\gamma = 0$ is $1.22$ and its corresponding critical values at levels $1\%, 5\%$, and $10\%$ are given in Table 7.1.1 as $-3.51, -2.89$, and $-2.58$ respectively.

| | Estimate | Std. Error | t value | Pr(>\|t\|) |
|---|---|---|---|---|
| (Intercept) | 1.42944 | 4.01643 | 0.35590 | 0.72323 |
| z.lag.1 | 0.01600 | 0.01307 | 1.22474 | 0.22571 |
| z.diff.lag | 0.36819 | 0.13440 | 2.73943 | 0.00820 |

Table 7: Regression with drift constant for the U.S. real GNP data.

From this analysis we conclude that the series behaves like a random walk with a drift constant term. The next question is whether further

| | 1pct | 5pct | 10pct |
|---|---|---|---|
| tau2 | -3.51 | -2.89 | -2.58 |
| phi1 | 6.70 | 4.71 | 3.86 |

Table 8: Dickey-Fuller critical values for test statistics with drift case.

differencing might be needed. So we simply repeat the unit root modelling and testing using the differenced series as input. The $\tau_3$ statistic equals to $-4.35$ which in comparison with the critical values given in Table 7.1.1 confirms that no further differencing is needed.

| | 1pct | 5pct | 10pct |
|---|---|---|---|
| tau3 | -4.04 | -3.45 | -3.15 |
| phi2 | 6.50 | 4.88 | 4.16 |
| phi3 | 8.73 | 6.49 | 5.47 |

Table 9: Dickey-Fuller critical values for test statistics for differencing case.

### *7.2. Covariate augmented tests*

The **CADFtest** package (Lupi, 2011) implements Hansen's covariate augmented Dickey-Fuller test (Hansen, 1995) by including stationary covariates in the model equations,

$$a(L)\Delta Z_t = \beta_0 + \beta_1 t + \gamma Z_{t-1} + b(L)'\Delta X_t + e_t \tag{10}$$

$$a(L)\Delta Z_t = \beta_0 + \gamma Z_{t-1} + b(L)'\Delta X_t + e_t, \tag{11}$$

$$a(L)\Delta Z_t = \gamma Z_{t-1} + b(L)'\Delta X_t + e_t. \tag{12}$$

where $a(L) = 1 - a_1 L + \ldots + a_p L^p$ and $b(L)' = b_{q2} L^{-q2} + \ldots + b_{q1} L^{q1}$. If the main function `CADFtest()` is applied without any stationary covariates, the ordinary ADF test is performed. In the illustrative example below, taken from the `CADFtest()` online documentation, the augmented test strongly rejects the unit root hypothesis, with a p-value less than 2%. When the covariate is not used, the unit root hypothesis is not rejected at the 5% level.

## 8. Cointegration and VAR models

In the simplest case, two difference-stationary time series are said to be cointegrated when a linear combination of them is stationary. Some classic examples (Engle and Granger, 1987) of this simple bivariate cointegrated series include:

- consumption and income
- wages and prices
- short and long term interest rates

Further examples are given in most time series textbooks with an emphasis on economic or financial series (Enders, 2010; Chan, 2010; Tsay, 2010; Lütkepohl, 2005; Hamilton, 1994; Banerjee et al., 1993). A cointegration analysis requires careful use of the methods discussed in these books since spurious relationships can easily be found when working with nonstationary or difference-stationary series (Granger and Newbold, 1974). Most financial and economic time series are not cointegrated. Cointegration implies a deep relationship between the series that is often of theoretical interest in economics. When a cointegrating relationship exists between two series, Granger causality must exist as well (Pfaff, 2006, p. 43). The **vars** package (Pfaff, 2010b) for vector autoregressive modelling is described in the book (Pfaff, 2006) and article (Pfaff, 2008). This package along its companion package **urca** (Pfaff, 2010a) provided modern comprehensive methods for cointegration analysis and modelling stationary and nonstationary multivariate time series. Full support for modelling, forecasting and analysis tools are provided for the vector autoregressive time series model (VAR), structural VAR (SVAR) and structural vector error-correction models (SVEC). The vector autoregressive, VAR $(p)$, stationary model for a $k$-dimensional time series $\{\boldsymbol{y}_t\}$

$$\boldsymbol{y}_t = \boldsymbol{\delta} \boldsymbol{d_t} + \boldsymbol{\Phi}_1 \boldsymbol{y}_{t-1} + \ldots + \boldsymbol{\Phi}_p \boldsymbol{y}_{t-p} + \boldsymbol{e}_t, \tag{13}$$

where $\boldsymbol{\delta}, \boldsymbol{\Phi}_\ell = (\phi_{ij,\ell})_{k\times k}$ are coefficient matrices, $\boldsymbol{d_t}$ is a matrix containing a constant term, linear trend, seasonal indicators or exogenous variables, and $\boldsymbol{\epsilon}_t \sim N(\boldsymbol{0}, I_k)$. The VAR model is estimated using OLS using the function `VARS` in the package **vars**. The basic VAR model without the covariates $\boldsymbol{d_t}$

may be estimated using the R core function `ar`. In the case of the SVAR model,

$$\boldsymbol{A}\boldsymbol{y}_t = \boldsymbol{\delta}\boldsymbol{d}_t + \boldsymbol{\Phi}_1\boldsymbol{y}_{t-1} + \ldots + \boldsymbol{\Phi}_p\boldsymbol{y}_{t-p} + \boldsymbol{B}\boldsymbol{e}_t, \tag{14}$$

where $\boldsymbol{A}$, and $\boldsymbol{B}$ are $k \times k$ matrices. With the structural models further restrictions are needed on the parameters and after the model has been uniquely specified, it is estimated by maximum likelihood. The SVEC model is useful for modelling non-stationary multivariate time series and is an essential tool in cointegration analysis. The basic error correction model, VEC, may be written,

$$\boldsymbol{\nabla}\boldsymbol{y}_t = \boldsymbol{\Pi}\boldsymbol{y}_t + \boldsymbol{\Gamma}_1\boldsymbol{\nabla}\boldsymbol{y}_{t-1} + \ldots + \boldsymbol{\nabla}\boldsymbol{\Gamma}_p\boldsymbol{y}_{t-p+1} + \boldsymbol{e}_t, \tag{15}$$

where $\boldsymbol{\nabla}$ is the first-differencing operator and $\boldsymbol{\Pi}$ and $\boldsymbol{\Gamma}_\ell, \ell = 1, \ldots, p-1$ are parameters. As with the VAR model the VEC model may be generalized to the SVEC model with coefficient matricies $\boldsymbol{A}$ and/or $\boldsymbol{B}$. A cointegration relationship exists provided that $0 < \text{rank } \boldsymbol{\Pi} < p$. When $\text{rank } \boldsymbol{\Pi} = 0$, a VAR model with the first differences may be used and when $\boldsymbol{\Pi}$ is of full rank, a model stationary VAR model of order $p$ is appropriate. The **vars** package includes functions for model fitting, model selection and diagnostic checking as well as forecasting with VAR, SVAR and SVEC models. Cointegration tests and analysis are provided in the **urca**. In addition to the two-step method of Engle and Granger (1987), tests based of the method of Phillips and Ouliaris (1990) and (Johansen, 1988; Johansen and Juselius, 1990) are implemented in the **urca** package. Illustrative examples of how to use the software for multivariate modelling and cointegration analysis are discussed in the book, paper and packages Pfaff (2006, 2008, 2010b)

## 9. GARCH time series

Volatility refers to the changes in variance or heteroscedasticity exhibited by many financial time series. The GARCH family of models (Engle, 1982; Bollerslev, 1986) capture quite well volatility clustering and the thick-tailed distributions often found with financial time series such as stock returns and foreign exchange rates. The GARCH family of models is discussed in more detail in textbooks dealing with financial time series (Enders, 2010; Chan, 2010; Tsay, 2010; Cryer and Chan, 2008; Shumway and Stoffer, 2011;

Hamilton, 1994). A GARCH$(p, q)$ sequence $a_t, t = \ldots, -1, 0, 1, \ldots$ is of the form

$$a_t = \sigma_t \epsilon_t$$

and

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{p} \alpha_j a_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2,$$

where $\alpha_0 > 0,\ \alpha_i \geq 0,\ 1 \leq i \leq p,\ \beta_j \geq 0,\ 1 \leq j \leq q$ are parameters. The errors $\epsilon_t$ are assumed to be independent and identically distributed from a parametric distribution such as normal, generalized error distribution (GED), Student-t or skewed variations of these distributions. While ARMA models deal with nonconstant conditional expectation, GARCH models handle non-constant conditional variance. Sometimes those two models are combined to form the ARMA/GARCH family of models. A comprehensive account of these models is also given in the book (Zivot and Wang, 2006) that also serves as the documentation for the well-known S-Plus add-on module, **Finmetrics**. Most of the functionality provided by **Finmetrics** for GARCH and related models is now available with the **fGARCH** package (Wuertz et al., 2009a). In the following, we give a brief discussion of the use **fGARCH** for simulation, fitting and inferences. The principal functions in this package include `garchSpec`, `garchSim`, and `garchFit` and related methods functions. The **fGarch** package allows for a variety distributional assumptions for the error sequence $\epsilon_t$. As an illustrative example, we simulate a GARCH(1,1) with $\alpha_0 = 10^{-6}$, $\alpha_1 = 0.2$, and $\beta_1 = 0.7$ and with a skewed GED distribution with skewness coefficient 1.25 and shape parameter 4.8. The simulated series is shown in Figure 22.

```
R> require("fGarch")
R> spec <- garchSpec(model = list(omega = 1e-06, alpha = 0.2,
+     beta = 0.7, skew = 1.25, shape = 4.8), cond.dist = "sged")
R> x <- garchSim(spec, n = 1000)
```

To fit the above simulated data with GARCH(1,1) we could use,

```
R> out = garchFit(~garch(1, 1), data = x, trace = FALSE)
```

Some of the inferences that can be carried out by using the `summary()` function, include the Jarque-Bera and Shapiro-Wilk normality tests, various Ljung-Box white noise tests, and ARCH effect tests. As a further

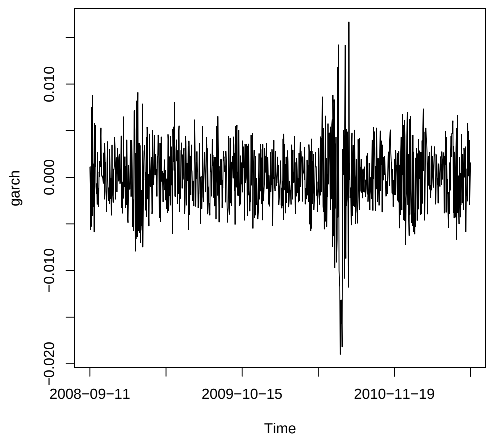


Figure 22: Simulated GARCH(1, 1) with $\alpha_0 = 10^{-6}, \alpha_1 = 0.2, \beta_1 = 0.7$.

illustration, we fit an ARMA/GARCH model to the U.S. inflation (Bollerslev, 1986). We used the GNP deflator for 1947-01-01 to 2010-04-01. There were $n = 254$ observations which are denoted by $z_t, t = 1, \ldots, n$. Then the inflation rate may be estimated by the logarithmic difference, $r_t = \log(z_t) - \log(z_{t-1})$.
The following ARMA/GARCH model was fit using the function `garchFit()` in **fGarch**, $r_t = 0.103 + 0.369 r_{t-1} + 0.223 r_{t-2} + 0.248 r_{t-3} + \epsilon_t$, and $\sigma_t^2 = 0.004 + 0.269 \epsilon_{t-1}^2 + 0.716 \sigma_{t-1}^2$. Figure 23 shows time series plots for $r_t$ and $\sigma_t$. The **tseries** (Trapletti, 2011) can also fit GARCH models but **fGarch** provides a much more comprehensive approach.

## 10. Wavelet methods in time series analysis

Consider a time series of dyadic length, $z_t, t = 1, \ldots, n$, where $n = 2^J$. The discrete wavelet transformation (DWT) decomposes the time series into $J$ wavelet coefficients vectors, $W_j, j = 0, \ldots, J-1$ each of length $n_j = 2^{J-j}, j = 1, \ldots, J$ plus a scaling coefficient $V_J$. Each wavelet coefficient coefficient is constructed as a difference of two weighted averages each of length $\lambda_j = 2^{j-1}$. Like the discrete Fourier transformation, the DWT provides an orthonormal decomposition, $W = \mathcal{W} Z$, where $W' = (W_1', \ldots, W_{J-1}', V_{J-1}')$, $Z = (z_1, \ldots, z_n)'$ and $\mathcal{W}$ is an orthonormal matrix.

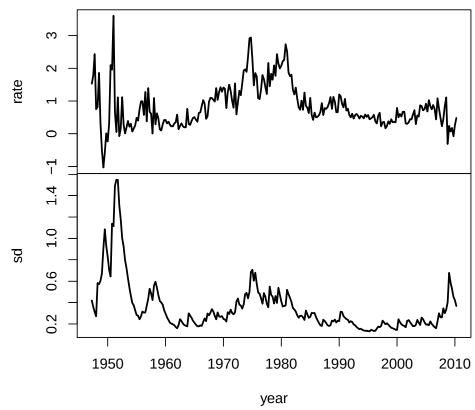


Figure 23: Inflation rate, $r_t$, and volatility, $\sigma_t$.

In practice, the DWT is not computed using matrix multiplication but much more efficiently using filtering and downsampling (Percival and Walden, 2000, Ch 4) The resulting algorithm is known as the pyramid algorithm and computationally it is even more efficient than the fast Fourier transform. Applying the operations in reverse order yields the inverse DWT. Sometimes a partial transformation is done, producing the wavelet coefficient vectors $W_j$, $j = 0, \ldots, J_0$, where $J_0 < J - 1$. In this case, the scaling coefficients are in the vector, $V_{J_0}$ of length $2^{J-J_0}$. The wavelet coefficients are associated with changes in the time series over scale $\lambda_j = 2^{j-1}$ while the scaling coefficients, $V_{J_0}$, are associated with the average level on scale $\tau = 2^{J_0}$. The maximum overlap DWT or MODWT omits the downsampling but it is still as fast as the fast Fourier transform. The MODWT has many advantages over the DWT (Percival and Walden, 2000, Ch 5) although it does not provide an orthogonal decomposition. Percival and Walden (2000) provide an extensive treatment of wavelet methods for time series research with many interesting scientific time series. Gençay et al. (2002) follows a similar approach to wavelets as given by Percival and Walden (2000) but with an emphasis on financial and economic applications. All important methods as well as all datasets discussed in the books by Percival and Walden (2000); Gençay et al. (2002) are available in the R packages `waveslim` (Whitcher, 2010) and `wmtsa` (?). Nason (2008) provides a general introduction to wavelet methods in statistics, including smoothing and multiscale time series analysis. R scripts are used extensively in his book and all figures in the book (Nason, 2008) may be reproduced using R scripts available in the `wavethresh` R package (Nason, 2010). Figure 24 shows the denoised annual Nile riverflows (Hipel and McLeod, 1994, p. 985) using the universal threshold with hard thresholding and Haar wavelets. Hipel and McLeod (1994); Hipel et al. (1975) fit a step intervention analysis time series model with AR(1) noise. Physical reasons as well as cumsum analysis were presented (Hipel and McLeod, 1994, §19.2.4) to suggest 1903 as the start of intervention that was due to the operation of the Aswan dam. The fitted step intervention is represented by the three line segments while the denoised flows are represented by the jagged curve. The points show actual observed flows. Figure 24 suggests the intervention actually may have started a few years prior to 1903. The computations for Figure 24 were done using the functions `modwt()`, `universal.thresh.modwt()` and `imodwt()` in the package `waveslim`. An estimate of the wavelet variance,

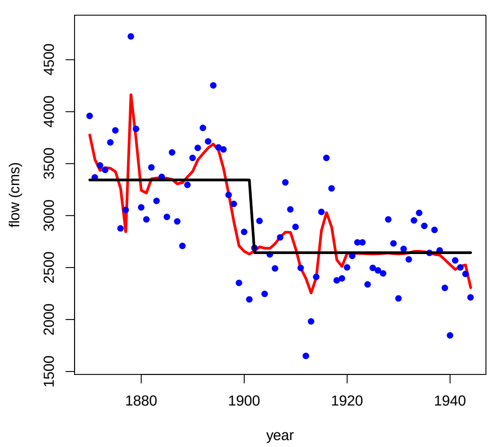


Figure 24: Mean annual Nile flow, October to September, Aswan.

$\hat{\sigma}^2(\lambda_j)$ is obtained based on the variance of the wavelet coefficients in an MODWT transformation at scale $\lambda_j = 2^{j-1}$. The wavelet variance is closely related to the power spectral density function and

$$\hat{\sigma}^2(\lambda_j) \approx 2 \int_{1/\lambda_j}^{2/\lambda_j} p(f)df.$$

The wavelet variance decomposition for the annual sunspot numbers, `sunspot.year` in R is shown in Figure 25. This figure was produced using the `wavVar` function in `wmtsa` and associated plot method. The 95% confidence intervals are shown in Figure 25. The wavelet variances correspond to changes over $1, 2, 4, 8$ and $16$ years. Multiresolution analysis

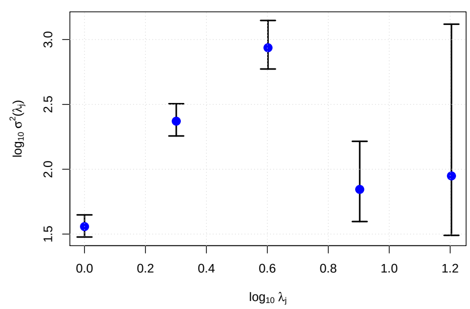


Figure 25: Wavelet variance, yearly sunspot numbers, 1700-1988

(MRA) is another widely useful wavelet method for time series analysis. The MRA decomposition works best with the MODWT. The `mra` function in `waveslim` was used to produce the decomposition of the shown in Figure 26. The `la8` or least-asymmetric filter with half-length $8$ was used (Percival and Walden, 2000, p. 109). A similar plot is given by Percival and Walden (2000, Figure 184).

## 11. Stochastic differential equations (SDE)

A SDE is comprised of a differential equation that includes a stochastic process, the simplest example being Brownian motion. The following SDE is often used to describe stock market prices, $dP(t) = P(t)\mu dt + \sigma dW(t)$, where $P(t)$ is the price at time $t$ and the parameters $\mu > 0$ and $\sigma > 0$ are the drift and diffusion parameters. The Gaussian white noise term, $W(t)$, may

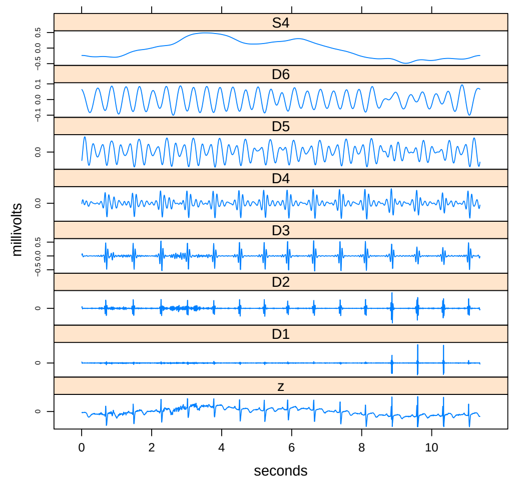


Figure 26: MRA using MODWT with `la8` filter. ECG time series comprised of about 15 beats of a human heart, sampled at 180 Hz, units are millivolts and $n = 2048$.

be considered the derivative of Brownian motion. More complicated SDE's may involve more complex drift and volatility functions. The book (Iacus, 2008) provides an intuitive and informal introduction to SDE and could be used in an introductory course on SDE. Only SDE's with Gaussian white noise are considered. The accompanying R package (Iacus, 2009) provides R scripts for all figures in the book (Iacus, 2008) as well as functions for simulation and statistical inference with SDE. An important area of application is in financial mathematics where option values or risk assessments are often driven by SDE systems. Usually Monte Carlo simulation is only way to find approximate solutions. The main class of SDE considered by this package is of the form of the following diffusion process

$$dX(t) = b(t, X(t))\, dt + \sigma(t, X(t))\, dW(t) \tag{16}$$

with some initial condition $X(0)$, where $W(t)$ is a standard Brownian motion. According to Itô formula, (16) can be represented as

$$X(t) = X(0) + \int_0^t b(u, X(u))\, du + \int_0^t \sigma(u, X(u))\, dW(u).$$

Under some regular conditions on the drift $b(\cdot, \cdot)$ and diffusion $\sigma^2(\cdot, \cdot)$, (16) has either a unique strong or weak solution. In practice, the class of SDE

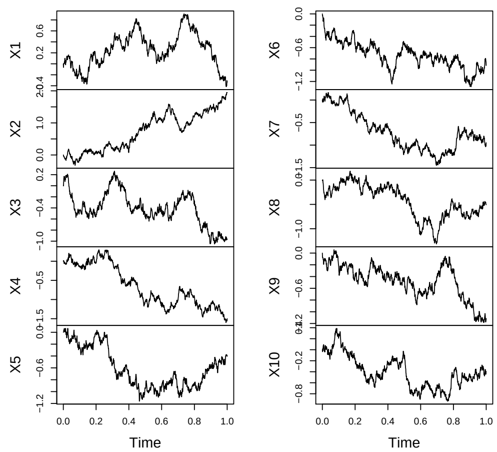


Figure 27: Ten Brownian Motions

given by (16) is too large. The following diffusion process covers many well-known and widely used stochastic processes, including Vasicek (VAS), Ornstein-Uhlenbeck (OU), Black-Scholes-Merton (BS) or geometric Brownian motion, and Cox-Ingersoll-Ross (CIR),

$$dX(t) = b(X(t))\,dt + \sigma(X(t))\,dW(t). \tag{17}$$

The main function is `sde.sim()` and it has extensive options for general diffusion process (17) or more specific processes. The function `DBride()` provides another general purpose function for simulating diffusion bridges. Simple to use functions for simulating a Brownian bridge and geometric Brownian motion, `BBridge()` and `GBM()`, are also provided. Using `sde.sim()`, we simulate ten replications of Brownian motions each starting at the $X(0) = 0$ and comprised of 1000 steps. The results are displayed in Figure 27. A more complex SDE,

$$dX(t) = (5 - 11x + 6x^2 - x^3)dt + dW(t)$$

with $X(0) = 5$ is simulated using three different algorithms and using two different step-sizes $\Delta = 0.1$ and $\Delta = 0.25$. For the smaller step size $\Delta = 0.1$, Figure 28 suggests all three algorithms work about equally well. But only the Shoji-Ozaki algorithm appears to work with the larger step size $\Delta = 0.25$. In addition to simulation, the `sde` package provides functions for

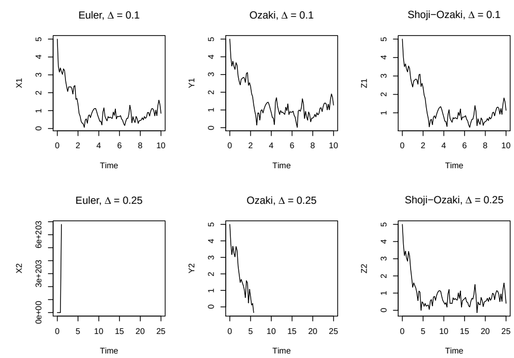


Figure 28: Simulations of $dX(t) = (5-11x+6x^2-x^3)dt+dW(t)$ using three different algorithms and two different step sizes.

parametric and nonparametric estimation: `EULERloglik()`, `ksmooth()`, `SIMloglik()`, and `simple.ef()`. Approximation of conditional density $X(t)|X(t_0) = x_0$ at point $x_0$ of a diffusion process is available with the functions: `dcElerian()`, `dcEuler()`, `dcKessler()`, `dcozaki()`, `dcShoji()`, and `dcSim()`.

## 12. Conclusion

There are many more packages available for time series than discussed in this article and many of these are briefly described in the CRAN Task Views, http://cran.r-project.org/web/views/. In particular, see task views for **Econometrics**, **Finance** and **TimeSeries**. We have tried to select the packages that might be of most general interest, that have been most widely used and that we are most familiar with. The reader should note that the packages published on CRAN, including those in the task views, need only obey formatting rules and not produce computer errors. There is absolutely no endorsement that packages available on CRAN produce valid or meaningful results. On the other, packages discussed in the *Journal of Statistical Software* or published by major publishers such as Springer-Verlag or Chapman & Hall/CRC have been carefully reviewed for correctness and quality. Researchers wishing to increase the impact of their work should consider implementing the results in R and making it available as a package on CRAN. Developing R packages is discussed in the online publication (R Development Core Team, 2011) and from a broader perspective by Chambers (2008). ,

## 13. Appendix

### *13.1.* **datasets**

| Dataset name | Description |
|---|---|
| `AirPassengers` | monthly airline passenger numbers 1949-1960 |
| `BJsales` | sales data with leading indicator |
| `BOD` | biochemical oxygen demand |
| `EuStockMarkets` | daily closing prices of major European stock indices, 1991-1998 |
| `LakeHuron` | level of Lake Huron 1875-1972 |
| `Nile` | flow of the river Nile |
| `Seatbelts` | road casualties in Great Britain 1969-84 |
| `UKDriverDeaths` | road casualties in Great Britain 1969-84 |
| `UKgas` | UK quarterly gas consumption |
| `USAccDeaths` | accidental deaths in the US 1973-1978 |
| `USPersonalExpenditure` | personal expenditure data |
| `WWWusage` | internet usage per minute |
| `WorldPhones` | the world's telephones |
| `airmiles` | passenger miles on commercial US airlines, 1937-1960 |
| `austres` | quarterly time series of the number of Australian residents |
| `co2` | Mauna Loa atmospheric co2 concentration |
| `UKLungDeaths` | monthly deaths from lung diseases in the UK |
| `freeny` | Freeny's revenue data |
| `longley` | Longley's economic regression data |
| `lynx` | annual Canadian lynx trappings 1821-1934 |
| `nhtemp` | average yearly temperatures in New Haven |
| `nottem` | average monthly temperatures at Nottingham, 1920-1939 |
| `sunspot.month` | monthly sunspot data, 1749-1997 |
| `sunspot.year` | yearly sunspot data, 1700-1988 |
| `sunspots` | monthly sunspot numbers, 1749-1983 |
| `treering` | yearly treering data, -6000-1979 |
| `uspop` | populations recorded by the US census |

Table 10: **datasets** time series data

| Function | Purpose |
|---|---|
| `embed` | matrix containing lagged values |
| `lag` | lagged values |
| `ts` | create a time series object |
| `ts.intersect` | form multivariate series by intersection |
| `ts.union` | form multivariate series by union |
| `time` | extract time from a ts-object |
| `cycle` | extract seasonal times from a ts-object |
| `frequency` | sampling interval |
| `window` | select subset of time series |

Table 11: **stats** utilities for ts-objects. These functions are useful for creating and manipulating univariate and multivariate time series.

| Function | Purpose |
|---|---|
| `acf` | acf, pacf |
| `ccf` | cross-correlation |
| `cpgram` | Bartlett's cumulate periodogram test |
| `lag.plot` | alternative time series plot |
| `fft` | fast Fourier transform |
| `convolve` | convolution via fft |
| `filter` | moving-average/autoregressive filtering |
| `spectrum` | periodic smoothing or autoregressive spectral density estimation |
| `toeplitz` | Toeplitz matrix |

Table 12: **stats** autocorrelation and spectral analysis functions

| Function | Purpose |
|---|---|
| `arima, arima0` | fit ARIMA |
| `ar` | fit AR |
| `KalmanLike` | loglikelihood for univariate state-space model |
| `KalmanRun` | filtering |
| `KalmanSmooth` | smoothing |
| `KalmanForecast` | forecasting |
| `makeARIMA` | arima to KF |
| `PP.test` | Phillips-Perron unit root test |
| `tsdiag` | diagnostic checks |
| `ARMAacf` | theoretical acf of ARMA |
| `acf2AR` | fit AR to acf |
| `Box.test` | Box-Pierce or Ljung-Box test |
| `diff, diffinv` | Box-Pierce or Ljung-Box test |
| `ARMAtoMA` | MA expansion for ARMA |
| `arima.sim` | simulate ARIMA |
| `HoltWinters` | Holt-Winters filtering |
| `StructTS` | moving-average/autoregressive filtering |

Table 13: **stats** functions for time series models. In addition many of these function have `predict` and `residuals` methods.

| Function | Purpose |
|---|---|
| `filter` | moving-average/autoregressive filtering |
| `tsSmooth` | smooth from StuctTS object |
| `stl` | seasonal-trend-loess decomposition |
| `decompose` | seasonal decomposition via moving-average filters |

Table 14: **stats** smoothing and filtering.

### *13.3.* **tseries**

| Function | Purpose |
|---|---|
| adf.test | augmented Dickey-Fuller test |
| bds.test | bds test |
| garch | fit garch models to time series |
| get.hist.quote | download historical finance data |
| jarque.bera.test | Jarque-Bera test |
| kpss.test KPSS | test for stationarity |
| quadmap | quadratic map (logistic equation) |
| runs.test | runs test |
| terasvirta.test | Teraesvirta neural network test for nonlinearity |
| tsbootstrap | bootstrap for general stationary data |
| white.test | White neural network test for nonlinearity |

Table 15: **tseries** functions

| Dataset name | Description |
|---|---|
| bev | Beveridge wheat price index, 1500-1869 |
| camp | Mount Campito yearly treering data, -3435-1969 |
| ice.river | Icelandic river Data |
| NelPlo | Nelson-Plosser macroeconomic time series |
| nino | sea surface temperature, nino 3 and nino 3.4 Indices |
| tcm | monthly yields on treasury securities |
| tcmd | daily yields on treasury securities |
| USeconomic | U.S. economic variables |

Table 16: **tseries** time series data

## References


Adler, J., 2009. R in a Nutshell. O'Reilly, Sebastopol, CA.

Aknouche, A., Bibi, A., 2009. Quasi-maximum likelihood estimation of periodic garch and periodic arma-garch processes. Journal of Time Series Analysis 30 (1), 19–46.

Baillie, R. T., 1996. Long memory processes and fractional integration in econometrics. Journal of Econometrics 73 (1), 5 – 59.

Banerjee, A., Dolado, J. J., Galbraith, J. W., Hendry, D. F., 1993. Cointegration, Error Correction, and the Econometric Analysis of Non-Stationary Data. Oxford University Press, Oxford.

Becker, R. A., Clark, L. A., Lambert, D., 1994. Cave plots: A graphical technique for comparing time series. Journal of Computational and Graphical Statistics 3 (3), 277–283.

Beran, J., 1994. Statistics for Long Memory Processes. Chapman & Hall/CRC, Boca Raton.

Beran, J., Whitcher, B., Maechler, M., 2009. longmemo: Statistics for Long-Memory Processes.
URL http://CRAN.R-project.org/package=longmemo

Bloomfield, P., 2000. Fourier Analysis of Time Series: An Introduction, 2nd Edition. Wiley, New York.

Bollerslev, T., 1986. Generalized autoregressive conditional heteroskedasticity. Journal of Econometrics 31 (3), 307–327.

Box, G., Jenkins, G. M., Reinsel, G. C., 2008. Time Series Analysis: Forecasting and Control, 4th Edition. Hoboken, N.J. : J.Wiley, New York.

Box, G. E. P., Tiao, G. C., 1975. Intervention analysis with applications to economic and environmental problems. Journal of the American Statistical Association 70 (349), 70–79.

Braun, W. J., Murdoch, D. J., 2008. A First Course in Statistical Programming with R. Cambridge University Press, Cambridge.

Brockwell, P. J., Davis, R. A., 1991. Time Series: Theory and Methods, 2nd Edition. Springer, New York.

Brown, R. L., Durbin, J., Evans, J. M., 1975. Techniques for testing the constancy of regression relationships over time. Journal of the Royal Statistical Society B 37, 149–163.

Chambers, J. M., June 2008. Software for Data Analysis: Programming with R. Statistics and Computing. Springer-Verlag.

Chan, K.-S., 2011. TSA: Time Series Analysis. R package version 0.98. URL http://CRAN.R-project.org/package=TSA

Chan, N. H., 2010. Time Series: Applications to Finance with R, 3rd Edition. Wiley, New York.

Cleveland, R. B., Cleveland, W. S., McRae, J. E., Terpenning, I., 1990. Stl: A seasonal-trend decomposition procedure based on loess. Journal of Official Statistics 6, 3–73.

Cleveland, W. S., 1993. Visualizing Data. Hobart Press.

Cleveland, W. S., McGill, M. E., McGill, R., 1988. The shape parameter of a two-variable graph. Journal of the American Statistical Association 83 (402), 289–300.

Cook, D., Swayne, D. F., 2007. Interactive Dynamic Graphics for Data Analysis R. Springer-Verlag, New York.

Cowpertwait, P. S., Metcalfe, A. V., 2009. Introductory Time Series with R. Springer Science+Business Media, LLC, New York.

Craigmile, P. F., 2003. Simulating a class of stationary gaussian processes using the davies-harte algorithm, with application to long memory processes. Journal of Time Series Analysis 24, 505–511.

Crawley, M. J., 2007. The R Book. Wiley, New York.

Cryer, J. D., Chan, K.-S., 2008. Time Series Analysis: With Applications in R, 2nd Edition. Springer Science+Business Media, LLC, New York.

Dalgaard, P., 2008. Introductory Statistics with R. Springer Science+Business Media, LLC, New York.

Davies, R. B., Harte, D. S., 1987. Tests for hurst effect. Biometrika 74, 95–101.

Dethlefsen, C., Lundbye-Christensen, S., Christensen, A. L., 2009. sspir: State Space Models in R.
URL http://CRAN.R-project.org/package=sspir

Diebold, F. X., Rudebusch, G. D., 1989. Long memory and persistence in aggregate output. Journal of Monetary Economics 24, 189–209.

Durbin, J., Koopman, S. J., 2001. Time Series Analysis by State Space Methods. Oxford University Press.

Elliott, G., Rothenberg, T. J., Stock, J. H., 1996. Efficient tests for an autoregressive unit root. Econometrica 64 (4), 813–836.

Enders, W., 2010. Applied Econometric Time Series, 3rd Edition. John Wiley and Sons, New York.

Engle, R. F., 1982. Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. Econometrica 50 (4), 987–1007.

Engle, R. F., Granger, C. W. J., 1987. Co-integration and error correction: Representation, estimation, and testing. Econometrica 55 (2), 251–276.

Everitt, B. S., Hothorn, T., 2009. A Handbook of Statistical Analyses Using R, 2nd Edition. Chapman and Hall/CRC, Boca Raton.

Faraway, J., Chatfield, C., 1998. Time series forecasting with neural networks: A comparative study using the airline data. Journal of the Royal Statistical Society. Series C (Applied Statistics) 47 (2), 231–250.

Fox, J., 2005. The R commander: A basic-statistics graphical user interface to R. Journal of Statistical Software 14 (9), 1–42.
URL http://www.jstatsoft.org/v14/i09

Fraley, C., Leisch, F., Maechler, M., Reisen, V., Lemonte, A., 2009. fracdiff: Fractionally differenced ARIMA aka ARFIMA(p,d,q) models.
URL http://CRAN.R-project.org/package=fracdiff

Friedman, J. H., 1991. Multivariate adaptive regression splines. The Annals of Statistics 19 (1), 1–67.

Gelper, S., Fried, R., Croux, C., 2010. Robust forecasting with exponential and holt-winters smoothing. Journal of Forecasting 29 (3), 285–300.

Gençay, R., Selçuk, F., Whitcher, B., 2002. An Introduction to Wavelets and Other Filtering Methods in Finance and Economics. Academic Press, New York.

Gentleman, R., 2009. R Programming for Bioinformatics. Chapman and Hall/CRC.

Gentleman, R., Ihaka, R., 1996. R: A language for data analysis and graphics. The Journal of Computational and Graphical Statistics 5 (2), 491–508.

Gilbert, P., 2011. dse: Dynamic Systems Estimation (time series package).
URL http://CRAN.R-project.org/package=dse

Granger, C. W. J., Newbold, P., 1974. Spurious regressions in econometrics. Journal of Econometrics 2, 111–120.

Granger, C. W. J., Newbold, P., 1976. Forecasting transformed series. Journal of the Royal Statistical Society. Series B (Methodological) 38 (2), 189–203.

Graves, S., 2011. FinTS: Companion to Tsay (2005) Analysis of Financial Time Series. R package version 0.4-4.
URL http://CRAN.R-project.org/package=FinTS

Grolemund, G., Wickham, H., 2011. Dates and times made easy with lubridate. Journal of Statistical Software 40 (3), 1–25.
URL http://www.jstatsoft.org/v40/i03

Hamilton, J. D., 1994. Time Series Analysis. Princeton University Press, Princeton, NJ.

Hansen, B. E., 1995. Rethinking the univariate approach to unit root testing: Using covariates to increase power. Econometric Theory 11 (5), 1148–1171.

Harrison, J., West, M., 1997. Bayesian forecasting and dynamic models.

Harvey, A., 1989. Forecasting, Structural Time Series Models and the Kalman Filter. Cambridge University Press, Cambridge.

Haslett, J., Raftery, A. E., 1989. Space-Time Modelling with Long-Memory Dependence: Assessing Ireland's Wind Power Resource. Journal of the Royal Statistical Society. Series C (Applied Statistics) 38 (1), 1–50.

Hastie, T., Tibshirani, R., 2011. mda: Mixture and Flexible Discriminant Analysis. R package version 0.4-2.
URL http://CRAN.R-project.org/package=mda

Hastie, T., Tibshirani, R., Friedman, J. H., 2009. The Elements of Statistical Learning, 2nd Edition. Springer-Verlag, New York.

Heiberger, R. M., Neuwirth, E., 2009. R Through Excel: A Spreadsheet Interface for Statistics, Data Analysis, and Graphics. Springer Science+Business Media, LLC, New York.

Helske, J., 2011. KFAS: Kalman filter and smoothers for exponential family state space models.
URL http://CRAN.R-project.org/package=KFAS

Hipel, K. W., Lennox, W. C., Unny, T. E., McLeod, A. I., 1975. Intervention analysis in water resources. Water Resources Research 11 (6), 855–861.

Hipel, K. W., McLeod, A. I., 1994. Time Series Modelling of Water Resources and Environmental Systems. Elsevier, Amsterdam, electronic reprint available, http://www.stats.uwo.ca/faculty/aim/1994Book/.

Hoffmann, T. J., 2011. Passing in command line arguments and parallel cluster/multicore batching in r with batch. Journal of Statistical Software, Code Snippets 39 (1), 1–11.
URL http://www.jstatsoft.org/v39/c01

Hopwood, W. S., McKeown, J. C., Newbold, P., 1984. Time series forecasting models involving power transformations. Journal of Forecasting 3 (1), 57–61.

Hothorn, T., Zeileis, A., Millo, G., Mitchell, D., 2010. lmtest: Testing Linear Regression Models. R package version 0.9-27.
URL http://CRAN.R-project.org/package=lmtest

Hyndman, R. J., 2010. forecast: Forecasting functions for time series. R package version 2.17.
URL http://CRAN.R-project.org/package=forecast

Hyndman, R. J., Khandakar, Y., 2008. Automatic time series forecasting: The forecast package for R. Journal of Statistical Software 27 (3), 1–22.
URL http://www.jstatsoft.org/v27/i03

Hyndman, R. J., Koehler, A. B., Ord, J. K., Snyder, R. D., 2008. Forecasting with Exponential Smoothing: The State Space Approach. Springer-Verlag, New York.

Iacus, S. M., 2008. Simulation and Inference for Stochastic Differential Equations: With R Examples. Springer Science+Business Media, LLC, New York.

Iacus, S. M., 2009. sde: Simulation and Inference for Stochastic Differential Equations. R package version 2.0.10.
URL http://CRAN.R-project.org/package=sde

Johansen, S., 1988. Statistical analysis of cointegration vectors. Journal of Economic Dynamics and Control 12, 231–254.

Johansen, S., Juselius, K., 1990. Maximum likelihood estimation and inference on cointegration-with applications to the demand for money. Oxford Bulletin of Economics and Statistics 52 (2), 169–210.

Kajitani, Y., Hipel, K. W., McLeod, A. I., 2005. Forecasting nonlinear time series with feed-forward neural networks: A case study of canadian lynx data. Journal of Forecasting 24, 105–117.

Kedem, B., Fokianos, K., 2002. Regression Models for Time Series Analysis. Wiley, New York.

Keenan, D. M., 1985. A Tukey nonadditivity-type test for time series nonlinearity. Biometrika 72, 39–44.

Kleiber, C., Zeileis, A., 2008. Applied Econometrics with R. Springer, New York.

Kwiatkowski, D., Phillips, P. C. B., Schmidt, P., Shin, Y., 1992. Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root? Journal of Econometrics 54, 159–178.

Lee, T. H., White, H., Granger, C. W. J., 1993. Testing for neglected nonlinearity in time series models. Journal of Econometrics 56, 269–290.

Leisch, F., 2002. Dynamic generation of statistical reports using literate data analysis. In: Härdle, W., Rönz, B. (Eds.), COMPSTAT 2002 – Proceedings in Computational Statistics. Physica-Verlag, Heidelberg, pp. 575–580.

Leisch, F., 2003. Sweave and beyond: Computations on text documents. In: Hornik, K., Leisch, F., Zeileis, A. (Eds.), Proceedings of the 3rd International Workshop on Distributed Statistical Computing, Vienna, Austria. ISSN 1609-395X.
URL
http://www.ci.tuwien.ac.at/Conferences/DSC-2003/Proceedings/

Lewis, P. A. W., Stevens, J. G., 1991. Nonlinear modeling of time series using multivariate adaptive regression splines (mars). Journal of the American Statistical Association 86 (416), 864–877.

Li, W. K., 1994. Time series models based on generalized linear models: Some further results. Biometrics 50 (2), 506–511.

Luethi, D., Erb, P., Otziger, S., 2010. FKF: Fast Kalman Filter.
URL http://CRAN.R-project.org/package=FKF

Lupi, C., 2011. CADFtest: Hansen's Covariate-Augmented Dickey-Fuller Test. R package version 0.3-1.
URL http://CRAN.R-project.org/package=CADFtest

Lütkepohl, H., 2005. New Introduction to Multiple Time Series Analysis. Springer-Verlag, New York.

Lütkepohl, H., Krätzig, M. (Eds.), 2004. Applied Time Series Econometrics. Cambridge University Press, Cambridge.

MacKinnon, J. G., 1996. Numerical distribution functions for unit root and cointegration tests. Journal of Applied Econometrics 11 (601-618).

McLeod, A., 1994. Diagnostic checking periodic autoregression models with application. The Journal of Time Series Analysis 15, 221–223, addendum, Journal of Time Series Analysis 16, p.647-648.

McLeod, A., 1998. Hyperbolic decay time series. The Journal of Time Series Analysis 19, 473–484.
URL http://www.jstatsoft.org/v23/i05

McLeod, A., Balcilar, M., 2011. pear: Package for Periodic Autoregression Analysis. R package version 1.2.
URL http://CRAN.R-project.org/package=pear

McLeod, A., Yu, H., Krougly, Z., 2007. Algorithms for linear time series analysis: With R package. Journal of Statistical Software 23 (5), 1–26.
URL http://www.jstatsoft.org/v23/i05

McLeod, A., Yu, H., Krougly, Z., 2011a. FGN: Fractional Gaussian Noise, estimation and simulation. R package version 1.4.
URL http://CRAN.R-project.org/package=ltsa

McLeod, A., Zhang, Y., 2008a. Faster arma maximum likelihood estimation. Computational Statistics and Data Analysis 52 (4), 2166–2176.

McLeod, A., Zhang, Y., 2008b. Improved subset autoregression: With R package. Journal of Statistical Software 28 (2), 1–28.
URL http://www.jstatsoft.org/v28/i02

McLeod, A., Zhang, Y., Xu, C., 2011b. FitAR: Subset AR Model Fitting. R package version 1.92.
URL http://CRAN.R-project.org/package=FitAR

McLeod, A. I., 2010. FitARMA: Fit ARMA or ARIMA Using Fast MLE Algorithm. R package version 1.4.
URL http://CRAN.R-project.org/package=FitARMA

McLeod, A. I., Li, W. K., 1983. Diagnostic checking arma time series models using squared-residual autocorrelations. Journal of Time Series Analysis 4, 269–273.

McLeod, A. I., Zhang, Y., 2006. Partial autocorrelation parameterization for subset autoregression. Journal of Time Series Analysis 27 (4), 599–612.

Meyer, D., June 2002. Naive time series forecasting methods: The holt-winters method in package ts. R News 2 (2), 7–10.

Milborrow, S., 2011. earth: Multivariate Adaptive Regression Spline Models. R package version 2.6-2.
URL http://CRAN.R-project.org/package=earth

Moore, D. S., 2007. The Basic Practice of Statistics, 4th Edition. W. H. Freeman & Co., New York.

Murrell, P., 2005. R Graphics. Chapman and Hall/CRC, Boca Raton.

Nason, G., 2008. Wavelet Methods in Statistics with R. Springer-Verlag, New York.

Nason, G., 2010. wavethresh: Wavelets statistics and transforms. R package version 4.5.
URL http://CRAN.R-project.org/package=wavethresh

Peng, R., 2008. A method for visualizing multivariate time series data. Journal of Statistical Software 25 (Code Snippet 1), 1–17.
URL http://www.jstatsoft.org/v25/c01

Percival, D. B., Walden, A. T., 1993. Spectral Analysis For Physical Applications. Cambridge University Press, Cambridge.

Percival, D. B., Walden, A. T., 2000. Wavelet Methods for Time Series Analysis. Cambridge University Press, Cambridge.

Petris, G., 2010. dlm: Bayesian and Likelihood Analysis of Dynamic Linear Models.
URL http://CRAN.R-project.org/package=dlm

Petris, G., Petrone, S., Campagnoli, P., 2009. Dynamic Linear Models with R. Springer Science+Business Media, LLC, New York.

Pfaff, B., 2006. Analysis of Integrated and Cointegrated Time Series with R. Springer, New York.

Pfaff, B., 2008. Var, svar and svec models: Implementation within R package vars. Journal of Statistical Software 27 (4), 1–32.
URL http://www.jstatsoft.org/v27/i04

Pfaff, B., 2010a. urca: Unit root and cointegration tests for time series data. R package version 1.2-5.
URL http://CRAN.R-project.org/package=urca

Pfaff, B., 2010b. vars: VAR Modelling. R package version 1.4-8.
URL http://CRAN.R-project.org/package=vars

Phillips, P. C. B., Ouliaris, S., 1990. Asymptotic properties of residual based tests for cointegration. Econometrica 58, 165–193.

Phillips, P. C. B., Perron, P., 1988. Testing for a unit root in time series regression. Biometrika 75 (2), 335–346.

R Development Core Team, 2011. Writing R Extensions. R Foundation for Statistical Computing, Vienna, Austria.
URL http://www.R-project.org/

Revolution Computing, 2011. foreach: For each looping construct for R. R package version 1.3.2.
URL http://CRAN.R-project.org/package=foreach

Ripley, B., 2011. nnet: Feed-forward Neural Networks and Multinomial Log-Linear Models. R package version 7.3-1.
URL http://CRAN.R-project.org/package=nnet

Ripley, B. D., 1996. Pattern Recognition and Neural Networks. Cambridge University Press, New York.

Ripley, B. D., June 2002. Time series in R 1.5.0. R News 2 (2), 2–7.

Ritz, C., Streibig, J. C., 2008. Nonlinear Regression with R. Springer Science+Business Media, LLC, New York.

Said, S. E., Dickey, D. A., 1984. Test for unit roots in autoregressive-moving average models of unknown order. Biometrika 71 (3), 599–607.

Sarkar, D., 2008. Lattice: Multivariate Data Visualization with R. Springer, New York.

Schmidberger, M., Morgan, M., Eddelbuettel, D., Yu, H., Tierney, L., Mansmann, U., 8 2009. State of the art in parallel computing with R. Journal of Statistical Software 31 (1), 1–27.
URL http://www.jstatsoft.org/v31/i01

Shadish, W. R., Cook, T. D., Campbell, D. T., 2001. Experimental and Quasi-Experimental Designs for Generalized Causal Inference, 2nd Edition. Houghton Mifflin.

Shumway, R. H., Stoffer, D. S., 2011. Time Series Analysis and Its Applications With R Examples, 3rd Edition. Springer.

Spector, P., 2008. Data Manipulation with R. Springer-Verlag, Berlin.

Teraesvirta, T., Lin, C. F., Granger, C. W. J., 1993. Power of the neural network linearity test. Journal of Time Series Analysis 14, 209–220.

Tesfaye, Y. G., Anderson, P. L., Meerschaert, M. M., 2011. Asymptotic results for fourier-parma time series. Journal of Time Series Analysis 32 (2), 157–174.

Thompson, M. E., McLeod, A. I., June 1976. The effects of economic variables upon the demand for cigarettes in Canada. The Mathematical Scientist 1, 121–132.

Trapletti, A., 2011. tseries: Time series analysis and computational finance. R package version 0.10-25.
URL http://CRAN.R-project.org/package=tseries

Tsay, R. S., 2010. Analysis of Financial Time Series, 3rd Edition. Wiley, New York.

Tusell, F., 2011. Kalman filtering in R. Journal of Statistical Software 39 (2).
URL http://www.jstatsoft.org/v39/i02

Ursu, E., Duchesne, P., 2009. On modelling and diagnostic checking of vector periodic autoregressive time series models. Journal of Time Series Analysis 30 (1), 70–96.

Venables, W. N., Ripley, B. D., 2000. S Programming. Springer.

Venables, W. N., Ripley, B. D., 2002. Modern Applied Statistics with S, 4th Edition. Springer.

Venables, W. N., Smith, D. M., R Development Core Team, 2011. Introduction to R. R Foundation for Statistical Computing, Vienna, Austria.
URL http://www.R-project.org/

Vingilis, E., McLeod, A. I., Seeley, J., Mann, R. E., Stoduto, G., Compton, C., Beirness, D., 2005. Road safety impact of extended drinking hours in ontario. Accident Analysis and Prevention 37, 547–556.

Whitcher, B., 2010. waveslim: Basic wavelet routines for one-, two- and three-dimensional signal processing. R package version 1.6.4.
URL http://CRAN.R-project.org/package=waveslim

Wickham, H., 2009. ggplot2: Elegant graphics for data analysis. Springer New York.
URL http://had.co.nz/ggplot2/book

Wilkinson, L., 1999. The Grammar of Graphics. Springer, New York.

Wolfram Research, I., 2011. Mathematica Edition: Version 8.0. Champaign, Illinois.

Wood, S., 2006. Generalized Additive Models: An Introduction with R. Chapman and Hall/CRC.

Wuertz, D., Chalabi, Y., 2011. timeSeries: Rmetrics - Financial Time Series Objects. R package version 2130.92.
URL http://CRAN.R-project.org/package=timeSeries

Wuertz, D., core team members, R., 2010. fBasics: Rmetrics - Markets and Basic Statistics. R package version 2110.79.
URL http://CRAN.R-project.org/package=fBasics

Wuertz, D., et al., 2009a. fGarch: Rmetrics - Autoregressive Conditional Heteroskedastic Modelling. R package version 2110.80.
URL http://CRAN.R-project.org/package=fGarch

Wuertz, D., et al., 2009b. fUnitRoots: Trends and Unit Roots. R package version 2100.76.
URL http://CRAN.R-project.org/package=fUnitRoots

Würtz, D., 2004. Rmetrics: An Environment for Teaching Financial Engineering and Computational Finance with R. Rmetrics, ITP, ETH Zürich, Swiss Federal Institute of Technology, ETH Zürich, Switzerland.
URL http://www.rmetrics.org

Zeileis, A., 2010. dynlm: Dynamic Linear Regression. R package version 0.3-0.
URL http://CRAN.R-project.org/package=dynlm

Zeileis, A., Leisch, F., Hansen, B., Hornik, K., Kleiber, C., 2010. strucchange: Testing, Monitoring and Dating Structural Changes. R package version 1.4-4.
URL http://CRAN.R-project.org/package=strucchange

Zeileis, A., Leisch, F., Hornik, K., Kleiber, C., 2002. Strucchange: An R package for testing for structural change in linear regression models. Journal of Statistical Software 7 (2), 1–38.
URL http://www.jstatsoft.org/v07/i02

Zhang, Y., McLeod, A. I., 2006. Computer algebra derivation of the bias of burg estimators. Journal of Time Series Analysis 27, 157–165.

Zhou, L., Braun, W. J., 2010. Fun with the r grid package. Journal of Statistics Education 18.
URL http://www.amstat.org/publications/jse/v18n3/zhou.pdf

Zivot, E., Wang, J., 2006. Modeling Financial Time Series with S-PLUS, 2nd Edition. Springer Science+Business Media, Inc, New York.

Zucchini, W., MacDonald, I. L., 2009. Hidden Markov Models for Time Series: A Practical Introduction using R, 2nd Edition. Chapman &Hall/CRC, Boca Raton.

Zuur, A. F., Ieno, E. N., Meesters, E., 2009. A Beginner's Guide to R. Springer Science+Business Media, LLC, New York.