

A Survey of Methods for Forecasting, Policy Analysis and Planning, with Examples in R

Joseph George Caldwell

8 July 2019

Copyright © 2019 Joseph George Caldwell. All rights reserved.
Posted at Internet website <http://www.foundationwebsite.org> .
May be copied or reposted for noncommercial use, with attribution to author and website.

Contents

A Survey of Methods for Forecasting, Policy Analysis and Planning, with Examples in R.....	2
1. Overview	2
2. Causal Inference (Causal Modeling and Analysis)	4
3. Stochastic Models	6
Time-Series Models.....	6
Univariate Univariable Models (ARIMA Models)	7
Univariate multivariable models (transfer-function models, ARMAX models).....	29
Multivariate models (VAR, VARMA and VARMAX models) ...	47
Special Topics (ARCH and GARCH Models; Kalman Filter; Bayesian Estimation; Structural Models; Econometric Models)	60
Microsimulation Models	73
4. Deterministic Models (and Primarily Deterministic Models) ...	74

Computable general equilibrium models (CGE models)	74
Population-based (demographic, regional, small-area-estimation) models.....	81
Cost-Benefit Models	83
Anticipation Surveys	84
Content Analysis.....	84
Appendix A. Additional Discussion of the Univariate Univariable ARIMA Model	84
References.....	93

A Survey of Methods for Forecasting, Policy Analysis and Planning, with Examples in R

1. Overview

This article presents examples of a variety of methods used for forecasting and policy analysis. The primary purpose is to show that the methods can be implemented using data and software that are freely available on the Internet.

The examples selected for discussion are basic examples of the most-used quantitative forecasting methods. For forecasting methods that are less-used, the discussion is limited to availability of R software, not to illustration of application of the methods.

The discussion of each example includes description of major steps leading to the development of the model, but it does not include discussion of the rationale for those steps. A thorough presentation of the model-development process is included in the References.

This article is not intended as a tutorial in R, or as a tutorial in the various forecasting methods discussed. The intention is to discuss the level of availability of free computer software programs (in R, and in GAMS for small models), to illustrate the level of detail of program output, and to provide an indication of the level of effort required to implement the various methods.

The following is an outline of the classes of models to be considered.

1. Causal Inference (Causal Modeling and Analysis)
 - a. Estimability and exogeneity
 - b. Experimental data vs. observational data
2. Stochastic Models
 - a. Time-series models
 - i. Univariate univariable models (ARIMA models)
 - ii. Univariate multivariable models (transfer-function models, ARMAX models)
 - iii. Multivariate models (VAR, VARMA and VARMAX models)
 - iv. Special topics
 1. ARCH and GARCH models
 2. Kalman filtering
 3. Bayesian estimation
 4. Structural models
 - b. Microsimulation models
3. Deterministic models (and primarily deterministic models)
 - a. Computable general equilibrium models (CGE models)
 - b. Population-based (demographic, regional, small-area estimation) models
 - c. Cost-benefit models
 - d. Anticipation surveys
 - e. Content analysis
4. References

The discussion of the first model, the univariate univariable model, is more detailed than the discussion of the later models. The examples do not present all of the R code required to implement the models, or to reproduce all of the output presented here. The purpose is to provide an indication of the level of detail output by typical available R software, and the level of effort required to implement it.

2. Causal Inference (Causal Modeling and Analysis)

Estimability and Exogeneity

When models include more than one variable, forecasts of a variable of interest – the dependent variable, or response variable – may be made conditional on assumed or predicted values for the other variables (the explanatory variables). The validity of such forecasts depends on the causal relationship of the dependent variable to the explanatory variables.

A “forecast” is an estimate of a future value of a variable. The key issue to be addressed in forecasting problems is determination of how to construct good forecasts (i.e., unbiased or consistent estimates having small standard errors). Over the past half-century, a considerable body of knowledge has developed on the theory and practice of forecasting based on statistical models. (Other terms that will be used in this article are “prediction” and “projection.” The terms “prediction” and “forecast” are interchangeable, and refer to an estimate of a future value based on a tested statistical model. A “projection” is simply the extrapolation of a curve or formula, such as an economic trend or population projection, without consideration of a tested (validated) mathematical model displaying the salient statistical properties of the process generating the phenomenon under study. A

“forecast” is a point or interval estimate of a quantity, along with an assessment of the likelihood of the estimate.)

A useful way of describing causal relationships is through the use of causal model diagrams, as described by Judea Pearl in his book, *Causality: Models, Reasoning, and Inference*, 2nd Edition, Cambridge University Press, 2009 (1st Ed. 2000). The causal model diagrams which Pearl uses to describe causal relationships are directed acyclic graphs (DAGs), or Bayesian networks. From a directed acyclic graph representation of a causal system, the validity (estimability) of conditional forecasts may be determined. This validity is assessed by means of criteria that are defined in terms of features of the DAG, such as the back-door criterion and the front-door criterion.

An alternative way of assessing the validity of conditional forecasts is to specify features of the joint probability distribution of the model variables. Under this approach, three different situations are described, called exogeneity conditions, relative to estimation of a specified parameter of interest.

The three main types of exogeneity are discussed in the book, *Co-integration, Error-Correction, and the Econometric Analysis of Non-Stationary Data* by Anindya Banerjee, Juan Dolado, John W. Galbraith, and David F. Hendry (Oxford University Press, 1993). These are:

Weak exogeneity. In this situation, forecasts may be made based on the conditional distribution, given the explanatory variables.

Strong exogeneity. In this situation, forecasts may be made conditional on forecasted future values of the explanatory variables.

Super exogeneity. In this situation, forecasts may be made conditional on specified future values of the explanatory variables.

While these exogeneity types can be defined in terms of theoretical factorization properties of the joint distribution, there is in general no practical way of assessing whether these conditions hold in practice. With the Pearl approach, once a causal model diagram is specified in terms of a DAG, the estimability criteria may be readily assessed from visual inspection of the DAG.

Experimental Data vs. Observational Data

If a forecasting model is estimated using data from a randomized experiment, the causal model is very simple. Specification of the levels of the explanatory variables is determined by randomization, and is therefore unrelated to any variables. The estimated model may be used to predict the effects of changes in the dependent variables. This is the situation in many laboratory experiments.

If a forecasting model is estimated from observational data (i.e., data that are passively observed, for which the levels of the explanatory variables are not determined by randomization), then the estimation must be based on a causal model, and the forecasts made in accordance with that model. This is the situation in many economic applications.

3. Stochastic Models

Time-Series Models

Most applications of forecasting are concerned with short-term forecasting. The term short-term refers to making predictions a

few (two or three) time periods into the future, where the time period refers to the standard reporting interval for the variable of interest. The length of the time interval can vary from very short, such as one-minute intervals in foreign-exchange trading, to months, quarters, or years.

The most widely used class of short-term forecasting models are time-series models, which are estimated from data collected at regularly spaced time intervals. In order to construct a model of reasonable accuracy, it is usually necessary to have available at least one hundred observations. The most widely used class of time-series models are autoregressive integrated moving average models, or ARIMA models. These models are also referred to as “Box-Jenkins” models, after the two statisticians who showed that this class of models could describe many real-world phenomena well, and popularized their use.

Box-Jenkins models may involve a single variable or multiple variables. Examples of Box-Jenkins models will now be presented.

Univariate Univariable Models (ARIMA Models)

A Nonseasonal Model

The term “variable” refers to a quantity of interest, stochastic or otherwise, such as time, temperature, price, or rate of inflation. The term “variate” refers to a random variable, that is, a real-valued function defined on a sample space (or, more accurately, a measurable function defined on a probability space). A univariate model involves a single random variable. A univariate model may or may not include other non-random variables. For example, a simple statistical regression model involves a single random

variable (the model error term) and a number of non-random variables (the explanatory variables), for example:

$$y_i = b_0 + b_1x_{1i} + b_2x_{2i} + e_i$$

where i denotes the observation number, y_i denotes the dependent (response, explained) variable, x_{1i} and x_{2i} denote explanatory (independent) variables, e_i is the model error term (a sequence of random variables having mean zero mean, uncorrelated with each other and with the x 's). (The dependent variable, y , is a random variable, but it is defined completely in terms of x_1 , x_2 and e , so that the model is defined in terms of a single random variable, not two.)

Suppose that z_t denotes the value of a time-related variable z at time t , and that t is specified as a sequence of equal-interval times, t_0, t_1, t_2, \dots . A standard model for describing the stochastic behavior of z_t is

$$z_t = \varphi_0 + \varphi_1z_{t-1} + \varphi_2z_{t-2} + \dots + \varphi_pz_{t-p} + a_t - \theta_1a_{t-1} - \theta_2a_{t-2} - \dots - \theta_qa_{t-q},$$

where

z_t = value of the observed random variable z at time t

a_t = model error term, a sequence of uncorrelated random variables with mean zero and variance σ^2

$\varphi_1, \varphi_2, \dots, \varphi_p$ = autoregressive parameters

$\theta_1, \theta_2, \dots, \theta_q$ = moving-average parameters.

If we introduce the backward-shift (or back-shift, or lag) operator, B , defined by $Bz_t = z_{t-1}$, then the preceding model may be written as

$$z_t = \varphi_0 + \varphi_1 B z_t + \dots + \varphi_p B^p z_t + a_t + \theta_1 B a_t + \dots + \theta_q B^q a_t$$

or

$$(1 - \varphi_1 B - \dots - \varphi_p B^p) z_t = (1 - \theta_1 B - \dots - \theta_q B^q) a_t$$

or

$$\Phi(B) z_t = \Theta(B) a_t$$

where

$$\Phi(B) = 1 - \varphi_1 B - \dots - \varphi_p B^p$$

and

$$\Theta(B) = 1 - \theta_1 B - \dots - \theta_q B^q.$$

The polynomial $\Phi(B)$ is called the autoregressive polynomial, and the polynomial $\Theta(B)$ is called the moving-average polynomial.

If the roots of the polynomial $\Phi(B)$ are outside the unit circle, then the process z_t is *stationary* (i.e., the joint distribution of any sequence $z_{t-r}, z_{t-r-1}, \dots, z_{t-r-s}$ is the same for all integral values of $r \geq 0$ and $s \geq 0$).

If some of the roots of $\Phi(B)$ are on the unit circle, then the process exhibits homogeneous nonstationary behavior – it wanders, as a random walk. For example, consider the model

$$\Phi(B)(1-B)^d z_t = \Theta(B) a_t,$$

where the order (highest power of B) of $\Phi(B)$ is p and the order of $\Theta(B)$ is q. Such a model is said to be an autoregressive integrated moving average (or ARIMA) process of order (p,d,q).

For practical time series, the roots of the moving-average polynomial are outside the unit circle. If this condition holds, the process is said to be *invertible*, and may be represented as

$$a_t = \Theta^{-1}(B) \Phi(B)(1-B)^d z_t .$$

An example of an ARIMA process is given by the time series of one-minute prices on the EUR/USD foreign-exchange market. These data are available from internet website HistData.com at <https://www.histdata.com/> . This particular dataset is 31,165 one-minute prices for EUR/USD for the month of April, 2019.

The steps involved in constructing an ARIMA model to represent these data are as follows:

- List the data
- Plot the data
- Plot the autocorrelation function (ACF) of the data
- If the ACF indicates nonstationarity, transform the data to a stationary variate
- Plot the ACF and partial autocorrelation function (PACF) of the stationary variate
- From the ACF and PACF, identify a structure for a tentative model (i.e., values of p and q)
- Estimate the model parameters (ϕ_s , θ_s and σ_a^2)
- Assess model adequacy (from ACF and σ_a^2 , and possibly other criteria, such as the Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC) or the Hannan-Quinn Information Criterion (HQC))
- Revise the model, if indicated by diagnostic tests

- Make forecasts using the tested model
- Document the model development process and the tested model

(Usage of the term “identification” and “identified” is not standard. In the field of econometrics, a model is “identified” if the parameters that define the model are estimable. In the field of time-series analysis, the process of specifying a model structure (such as the values of p , d and q in an ARIMA model) is referred to as “identification.”)

These steps will now be summarized, for the EUR/USD data. The data will be analyzed using software programs available from the Comprehensive R Analysis Network, CRAN, at Internet website <https://cran.r-project.org/> , in the “stats” library of procedures.

Figure 1 presents a listing of the first 240 observations of the data set.

Figure 1. Listing of first 240 one-minute prices of EUR/USD forex data for April, 2019

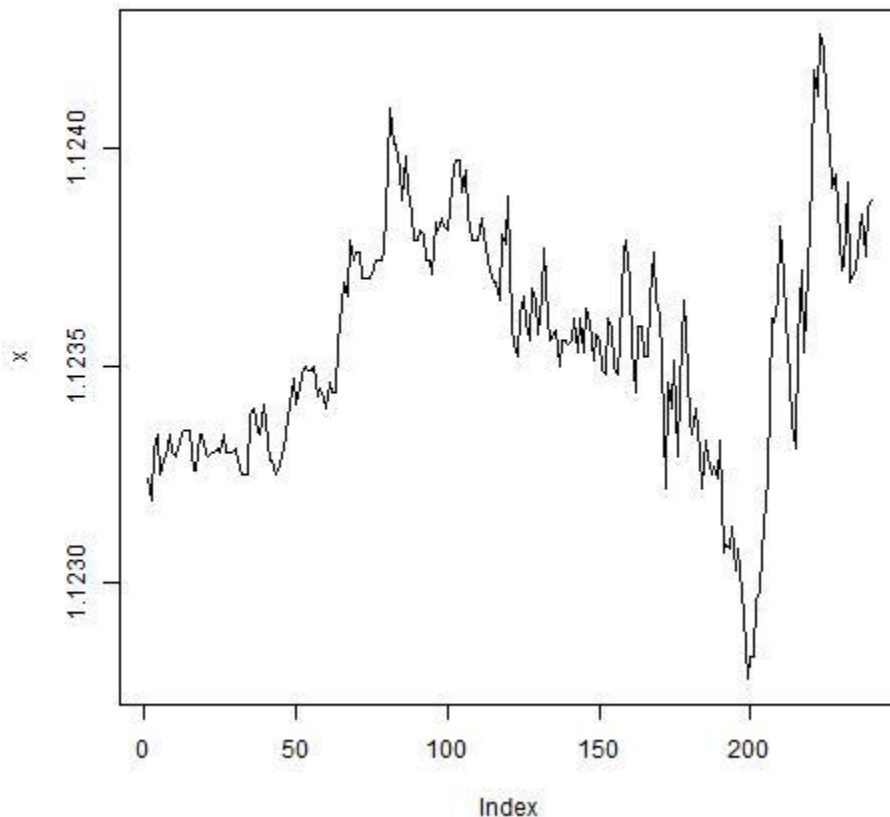
```
> print(x)
[1] 1.12324 1.12319 1.12331 1.12334 1.12325 1.12328 1.12329
1.12334 1.12330
[10] 1.12329 1.12331 1.12334 1.12335 1.12335 1.12335 1.12326
1.12326 1.12334
[19] 1.12334 1.12330 1.12329 1.12330 1.12330 1.12331 1.12330
1.12334 1.12330
[28] 1.12330 1.12330 1.12331 1.12328 1.12325 1.12325 1.12325
1.12339 1.12340
[37] 1.12335 1.12334 1.12341 1.12337 1.12328 1.12328 1.12325
1.12326 1.12329
[46] 1.12331 1.12338 1.12341 1.12347 1.12341 1.12345 1.12348
1.12350 1.12349
```

[55] 1.12349 1.12350 1.12343 1.12345 1.12343 1.12340 1.12346
1.12344 1.12344
[64] 1.12355 1.12364 1.12369 1.12366 1.12379 1.12374 1.12376
1.12376 1.12370
[73] 1.12370 1.12370 1.12371 1.12374 1.12374 1.12374 1.12376
1.12396 1.12409
[82] 1.12401 1.12401 1.12395 1.12388 1.12398 1.12391 1.12385
1.12379 1.12379
[91] 1.12381 1.12380 1.12374 1.12374 1.12371 1.12383 1.12381
1.12384 1.12382
[100] 1.12381 1.12386 1.12396 1.12397 1.12397 1.12390
1.12395 1.12385 1.12379
[109] 1.12379 1.12379 1.12384 1.12380 1.12374 1.12372
1.12369 1.12369 1.12365
[118] 1.12380 1.12378 1.12389 1.12359 1.12355 1.12352
1.12362 1.12366 1.12360
[127] 1.12356 1.12368 1.12365 1.12357 1.12366 1.12377
1.12361 1.12356 1.12357
[136] 1.12358 1.12350 1.12356 1.12356 1.12355 1.12356
1.12361 1.12353 1.12361
[145] 1.12353 1.12363 1.12360 1.12351 1.12357 1.12356
1.12349 1.12348 1.12361
[154] 1.12359 1.12350 1.12348 1.12356 1.12376 1.12379
1.12369 1.12354 1.12344
[163] 1.12359 1.12359 1.12352 1.12352 1.12365 1.12376
1.12365 1.12362 1.12351
[172] 1.12322 1.12346 1.12340 1.12351 1.12329 1.12345
1.12365 1.12358 1.12338
[181] 1.12334 1.12340 1.12330 1.12322 1.12333 1.12329
1.12325 1.12327 1.12324
[190] 1.12333 1.12307 1.12309 1.12308 1.12313 1.12303
1.12308 1.12298 1.12293
[199] 1.12278 1.12283 1.12283 1.12296 1.12298 1.12307
1.12319 1.12324 1.12361

[208] 1.12360 1.12365 1.12382 1.12371 1.12361 1.12347
1.12337 1.12331 1.12356
[217] 1.12372 1.12353 1.12372 1.12393 1.12418 1.12412
1.12426 1.12423 1.12412
[226] 1.12401 1.12391 1.12394 1.12386 1.12372 1.12373
1.12392 1.12369 1.12371
[235] 1.12372 1.12381 1.12385 1.12375 1.12386 1.12388

Figure 2 presents a time-series plot of those observations.

Figure 2. Plot of 240 One-Minute EUR/USA Foreign Exchange Prices



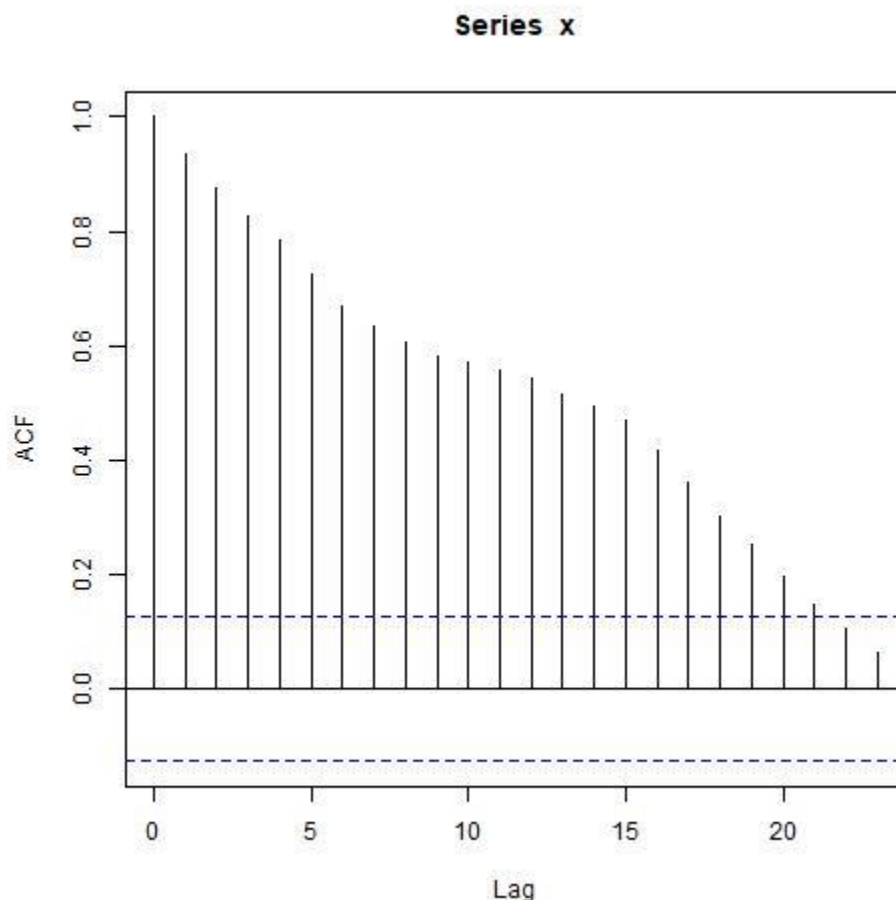
A sample size of 240 observations is plenty for which to estimate a Box-Jenkins model. For the entire month of April, 2019, there

are 31,665 observations. A quick visual scan of the month's data show that the first 240 observations appear to be typical.

The time series data were labeled as the variable "x". The R command for the data listing is `print(x)` and the R command for the plot is `plot(x, type="l")`. (Note: The analysis was performed using a source command file. For console entry of commands, the command "x" would have produced the listing. When using a command file, it is necessary to use the print function, "print(x)".)

Figure 3 shows a plot of the autocorrelation function (ACF) of the raw data. The R command for constructing the ACF is `acf(x)`.

Figure 3. Autocorrelation Function of Forex Price Data



The ACF dies out slowly, indicating nonstationary behavior, which is evident from the graph. Given the “wandering” appearance of the graph, and the slow decline of the ACF, it is reasonable to transform the raw data by taking first differences, i.e., by forming the transformed variate

$$z_t = (1 - B)x_t = x_t - x_{t-1}$$

where x_t denotes the original variable and z_t denotes the first difference. Figure 4 shows the ACF of the first differences.

Figure 4. Autocorrelation Function of First Differences of One-Minute EUR/USD Forex Price Data

```
> dx<-diff(x, lag = 1)

> print(dx[1:10])
[1] -0.00005 0.00012 0.00003 -0.00009 0.00003 0.00001
0.00005 -0.00004
[9] -0.00001 0.00002

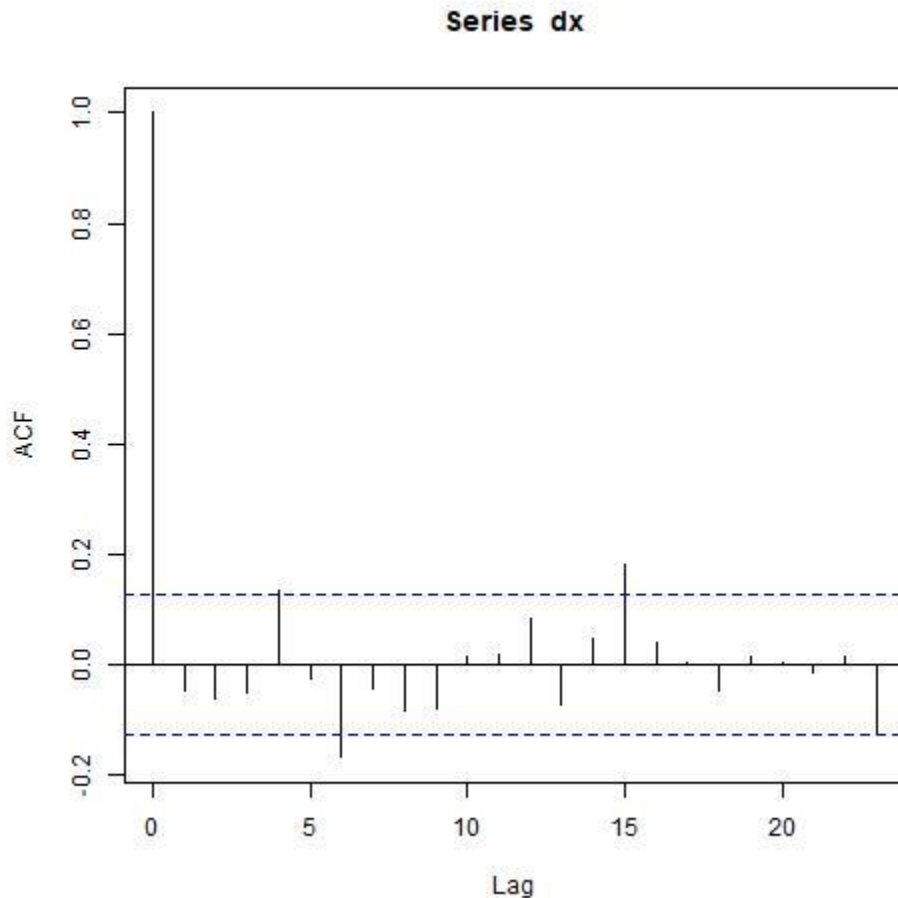
> print(summary(dx))
   Min.   1st Qu.   Median     Mean   3rd Qu.    Max.
-3.000e-04 -5.000e-05  0.000e+00  2.678e-06  5.000e-05
 3.700e-04

> print(acf(dx))
```

Autocorrelations of series ‘dx’, by lag

0	1	2	3	4	5	6	7	8	9	10
1.000	-0.047	-0.062	-0.053	0.132	-0.026	-0.168	-0.043	-0.083	-0.081	0.016
11	12	13	14	15	16	17	18	19	20	21

0.016 0.085 -0.075 0.048 0.180 0.041 0.003 -0.046 0.013
 0.005 -0.016
 22 23
 0.013 -0.123



The ACF of the first differences is essentially zero. The autocorrelation at lag 1 is small, -.047. This suggests that a reasonable model might be

$$Z_t = a_t$$

or

$$z_t = a_t - .047 a_{t-1}$$

or, in terms of the original variate x ,

$$x_t = x_{t-1} + a_t - .047a_{t-1}.$$

From the Comprehensive R Analysis Network, CRAN, at Internet website <https://cran.r-project.org/>, a program, `arima`, is available in the stats library, for estimating parameters of a univariate ARIMA model. The call to this program is

```
arima(x, order=c(p,d,q)).
```

From the preliminary data analysis, reasonable values of the parameters are $p = 0$, $d = 1$, and $q = 1$. Execution of the command

```
arima(x, order=c(0,1,1))
```

produces the output shown in Figure 5.

Figure 5. Output for ARIMA Model $(p,d,q) = (0,1,1)$

```
> model2<-arima(x, order=c(0,1,1))
```

```
> print(coef(model2))
```

```
      ma1  
-0.05336126
```

```
> print(vcov(model2))
```

```
      ma1  
ma1 0.004805707
```

```
> print(model2$sigma2)
```

```
[1] 8.439229e-09
```

The estimated model parameters are $\theta_1 = -.05336$ and $\sigma_a^2 = 8.439229e-09$. The estimated model is

$$x_t = x_{t-1} + a_t - .053a_{t-1}$$

with $\text{var}(a_t) = 8.439229e-09$.

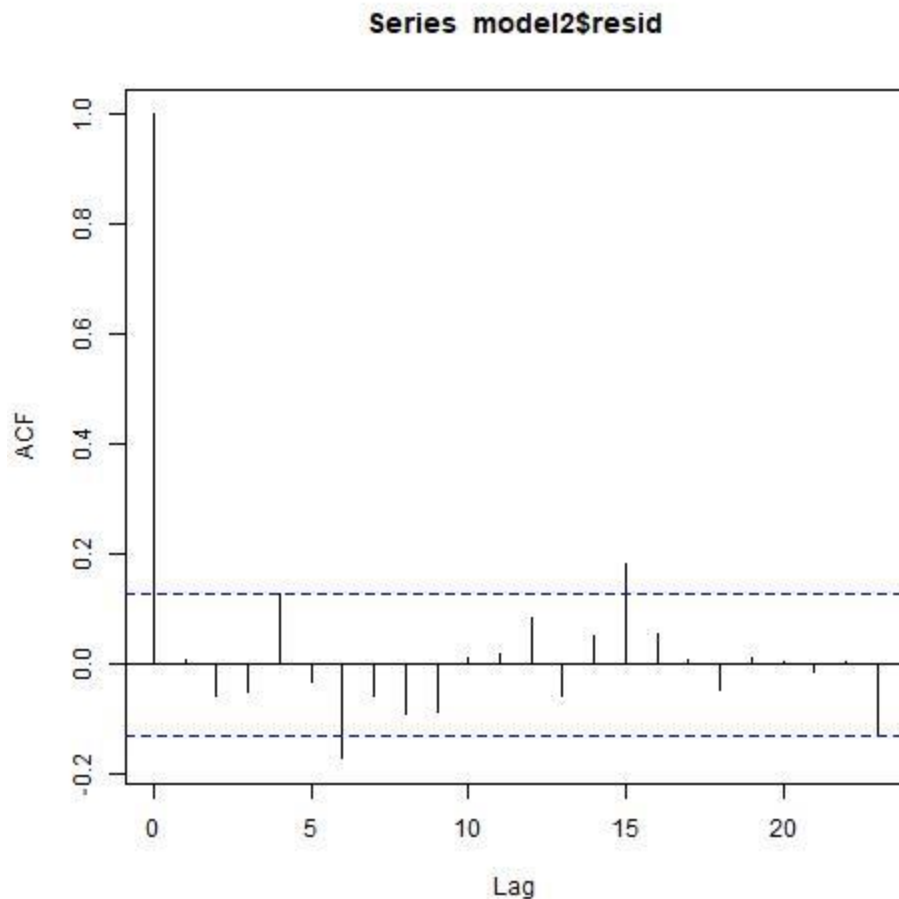
Figure 6 is a plot of the ACF of the model residuals.

Figure 6. Autocorrelation Function of ARIMA (p,d,q) = (0,1,1)
Model Residuals

```
> print(acf(model2$resid))
```

Autocorrelations of series 'model2\$resid', by lag

0	1	2	3	4	5	6	7	8	9	10
1.000	0.007	-0.059	-0.051	0.127	-0.032	-0.172	-0.059	-0.091	-	
0.086	0.011									
11	12	13	14	15	16	17	18	19	20	21
0.017	0.085	-0.059	0.053	0.182	0.054	0.008	-0.049	0.011		
0.004	-0.014									
22	23									
0.003	-0.132									



The ACF is essentially zero, indicating that the model is an adequate representation of the process.

For very simple models, such as the present one, a decision about model adequacy is generally based simply on a subjective assessment about the variance of the model residuals (smaller is preferred), the “whiteness” of the model residuals (i.e., visual flatness of the ACF, or results of a Box-Leung test) and the number of parameters (smaller is preferred), without consideration of information criteria such as the Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC) or the Hannan-Quinn Information Criterion (HQC)). Little additional effort is required to examine the information criteria, since they are generally computed automatically by ARIMA

estimation software. For more complex models, the information criteria and other criteria are applied to suggest model structure and compare alternative models. Simulation studies show have demonstrated the ability of the information criteria to correctly identify model structure. Even for small models, however, it is standard procedure to assess features such as stationarity, invertibility, the “flatness” of the error sum of squares (likelihood function assuming that the model error term is normally distributed), and presence or absence of deterministic or random periodicities. The primary purpose of this paper is to assess the availability of R routines for a variety of forecasting models and the effort involved in applying those models. It is not intended as a tutorial in the development of mathematical forecasting models, and so little discussion of the various model adequacy is presented. The cited references provide detailed treatment of these aspects of model development.

The model is essentially a random walk. For a random walk, the forecast is the most recent observed value of the series. (Note that the various programs available to estimate ARIMA models are not consistent in the signs that they use for the ϕ s and θ s in the specification of the ARIMA model. The R routine `arima` uses pluses on the θ terms, at difference with the specification used above (and by the Box-Jenkins book). In what follows, we shall use the Box-Jenkins notation, which implies $\theta_1 = .053$).

To construct forecasts, the following command is executed:

The program output is shown in Figure 6.

Figure 6. 12-step-ahead Forecast from Time $t=240$ for Estimated ARIMA Model

```
> predict(model2, n.ahead=12)
$pred
```

Time Series:

Start = 241

End = 252

Frequency = 1

[1] 1.123879 1.123879 1.123879 1.123879 1.123879 1.123879
1.123879 1.123879
[9] 1.123879 1.123879 1.123879 1.123879

\$se

Time Series:

Start = 241

End = 252

Frequency = 1

[1] 9.186528e-05 1.264984e-04 1.535071e-04 1.764286e-04
1.966968e-04
[6] 2.150634e-04 2.319803e-04 2.477447e-04 2.625644e-04
2.765911e-04
[11] 2.899401e-04 3.027010e-04

It is seen that the forecast is essentially the current value, modified slightly because of the nonzero value of θ_1 . The standard errors of the forecasts are:

Since the model is essentially a random walk, there is little information from the forecast that could be used to make profitable foreign-exchange trades.

Some additional discussion of the univariate univariable ARIMA model is presented in Appendix A, including discussion of the forecast function and a formula for estimating the variance of the forecast errors.

A Seasonal Model

The nonseasonal ARIMA model discussed above may be extended to include efficient representations of time series that exhibit seasonal behavior. A much-cited example of a seasonal time series model is the “airline ticket sales” example presented in Box and Jenkins’ book and in many later publications.

A data set for the airline ticket sales example is provided on the CRAN R website. The dataset is named **.

Figure7 presents a listing of the airline ticket sales data. The data set are logarithms of monthly international airline ticket sales.

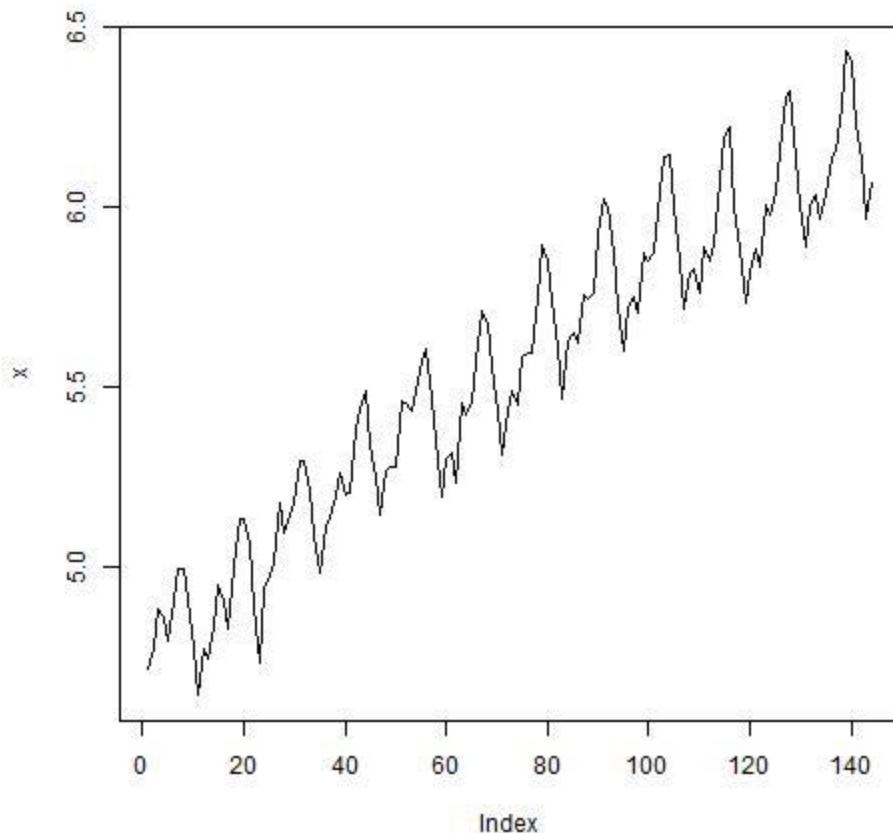
Figure 7. The Box-Jenkins Monthly International Ticket Sales Data Set

```
[1] 4.718499 4.770685 4.882802 4.859812 4.795791 4.905275
4.997212 4.997212
[9] 4.912655 4.779123 4.644391 4.770685 4.744932 4.836282
4.948760 4.905275
[17] 4.828314 5.003946 5.135798 5.135798 5.062595 4.890349
4.736198 4.941642
[25] 4.976734 5.010635 5.181784 5.093750 5.147494 5.181784
5.293305 5.293305
[33] 5.214936 5.087596 4.983607 5.111988 5.141664 5.192957
5.262690 5.198497
[41] 5.209486 5.384495 5.438079 5.488938 5.342334 5.252274
5.147494 5.267858
[49] 5.278115 5.278115 5.463832 5.459586 5.433722 5.493062
5.575949 5.605802
[57] 5.468060 5.351858 5.192957 5.303305 5.318120 5.236442
5.459586 5.424950
[65] 5.455321 5.575949 5.710427 5.680172 5.556828 5.433722
5.313206 5.433722
```

[73] 5.488938 5.451038 5.587249 5.594711 5.598422 5.752573
5.897154 5.849325
[81] 5.743003 5.613128 5.468060 5.627621 5.648974 5.624018
5.758902 5.746203
[89] 5.762052 5.924256 6.023448 6.003887 5.872118 5.723585
5.602119 5.723585
[97] 5.752573 5.707110 5.874931 5.852202 5.872118 6.045005
6.142037 6.146329
[105] 6.001415 5.849325 5.720312 5.817111 5.828946 5.762052
5.891644 5.852202
[113] 5.894403 6.075346 6.196444 6.224558 6.001415 5.883322
5.736572 5.820083
[121] 5.886104 5.834811 6.006353 5.981414 6.040255 6.156979
6.306275 6.326149
[129] 6.137727 6.008813 5.891644 6.003887 6.033086 5.968708
6.037871 6.133398
[137] 6.156979 6.282267 6.432940 6.406880 6.230482 6.133398
5.966147 6.068426

Figure 8 presents a time-series plot of the data.

Figure 8. Plot of International Ticket Sales Data



The plot shows clearly that the series is strongly seasonal and nonstationary. Transformation to a stationary variate can be accomplished for this time series by differencing. Because of the strong 12-month seasonal period, the differences of interest are the first difference, $D1 = (1 - B)$, the twelfth difference, $D12 = (1 - B^{12})$, and the concatenation of the first and twelfth differences, $D1D12 = (1 - B)(1 - B^{12})$. The autocorrelations of the three transformed variates obtained by applying those difference operators are shown in Figures 9, 10 and 11.

Figure 9. Autocorrelation Function of First-Differenced Series

Figure 10. Autocorrelation Function of Twelfth-Differenced Series

Figure 11. Autocorrelation Function of First- and Twelfth-Differenced Series

The ACF of the first- and twelfth-differenced series,

$$z_t = (1 - B)(1 - B^{12})x_t$$

Where x_t denotes the original series (in logarithms), exhibits stationary behavior (i.e., it dies out quickly). It has substantial autocorrelations for lags 1 and 13, suggesting that a model of the form

$$z_t = (1 - \theta_1)(1 - \theta_{12})a_t$$

Is a reasonable model structure to consider.

The same R procedure `arima` as was used earlier is used to estimate the model parameters. The procedure call is `arima(V1, order=c(0,1,1), seasonal=list(order = c(0,1,1), period=12))`, and the program model output is as follows:

```
> model1<-arima(V1, order=c(0,1,1), seasonal=list(order =  
c(0,1,1), period=12))
```

```
> print(coef(model1))  
      ma1      sma1  
-0.4018279 -0.5569452
```

```
> print(vcov(model1))  
      ma1      sma1  
ma1  0.0080359992 -0.0007254433  
sma1 -0.0007254433  0.0053435583
```

```
> print(model1$sigma2)
[1] 0.001348034
```

The estimates of the model parameters are $\theta_1 = .40$, $\theta_{12} = .56$ and $\sigma_a^2 = .00135$.

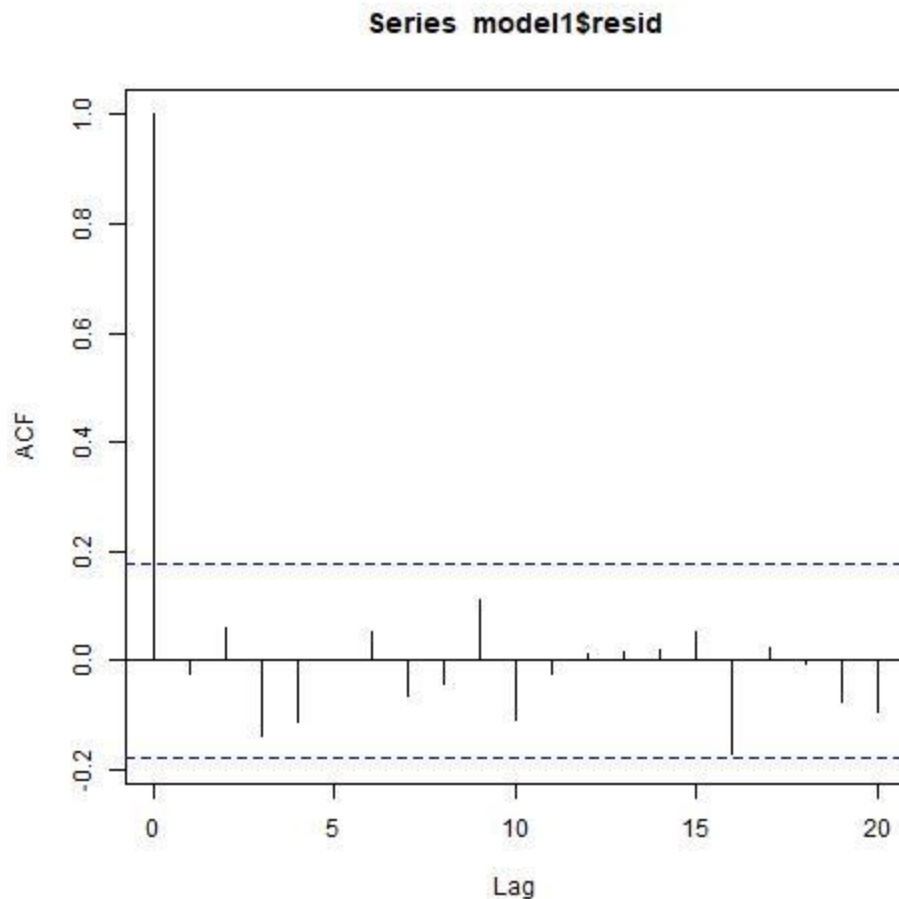
The ACF of the model residuals is

```
> print(acf(model1$resid))
```

Autocorrelations of series 'model1\$resid', by lag

0	1	2	3	4	5	6	7	8	9	10
1.000	-0.024	0.060	-0.140	-0.113	0.002	0.052	-0.065	-0.043	0.111	-0.107
11	12	13	14	15	16	17	18	19	20	
-0.024	0.013	0.018	0.019	0.054	-0.170	0.023	-0.008	-0.076	-0.095	

A plot of the ACF is shown in Figure 12.



The ACF of the model residuals indicates that they are a white noise series, so that the model is reasonable. Hence we see that it was possible to obtain a reasonable model representation of this seasonal time series using but two parameters (besides σ^2).

Forecasts for the model from the last observation, extending 24 time-steps into the future, are obtained by the procedure call `predict(model1, n.ahead=24)`, which produces the following program output (forecast and forecast standard errors):

```
> predict(model1, n.ahead=24)
$pred
Time Series:
Start = 145
```

End = 168

Frequency = 1

[1] 6.110186 6.053776 6.171715 6.199301 6.232557 6.368779
6.507294 6.502907

[9] 6.324699 6.209008 6.063488 6.168025 6.206436 6.150025
6.267965 6.295550

[17] 6.328806 6.465029 6.603543 6.599156 6.420948 6.305258
6.159737 6.264275

\$se

Time Series:

Start = 145

End = 168

Frequency = 1

[1] 0.03671561 0.04278290 0.04809071 0.05286830 0.05724856
0.06131670

[7] 0.06513123 0.06873440 0.07215787 0.07542611 0.07855850
0.08157069

[13] 0.09008474 0.09549706 0.10061867 0.10549193
0.11014979 0.11461852

[19] 0.11891944 0.12307015 0.12708537 0.13097755
0.13475737 0.13843402

Summary of Univariate Univariable Models

The class of univariate univariable models is able to describe the stochastic behavior of a wide range of real-world processes. Application of the Box-Jenkins' iterative approach of model specification, diagnosis, and re-specification until a satisfactory model is found requires observations on a single variable of interest for at least 100 equally spaced time intervals, preferably 200-400. The process of model development requires knowledge of the Box-Jenkins methodology and on the order of an hour of time for data processing, statistical analysis of the raw data and

alternative model specifications, and documentation of the model development process and the final model.

Computer software for performing the data analysis is available in a number of commercially available statistical program packages, such as Stata, SAS, and SPSS, in in the free R statistical software suite.

Univariate univariable models are easy to construct, but the fact that they involve only the response variable and no explanatory variables limits their usefulness. In financial applications, such as the forex price model developed above, many of the models are random walks or near-random-walks.

Univariate multivariable models (transfer-function models, ARMAX models)

In many applications, variables are present that have an influence on a variable of interest. Taking such variables into account can increase the forecasting precision. The variable of interest (to be forecast) is called by a variety of names, including the dependent variable, response variable, explained variable, output variable or controlled variable. The variable that has an influence on the dependent variable is called the independent variable, explanatory variable, input variable, or control variable. The independent variable may be a deterministic variable for which the level may be set, or it may be a random variable.

The situation we consider here is the one in which the model error term is uncorrelated with the explanatory variable. In terms of a causal model, the explanatory variable may have an effect on the response variable, but the response variable has no effect on the explanatory variable. The explanatory variable is said to be an exogenous variable (with respect to the dependent variable).

The explanatory variable may be controlled or passively observed. In a laboratory setting, it may be feasible to completely control the values of the input variable, such as in the situation in which a white noise signal is input to a filter to estimate the filter response. In economic applications, the input variable may be controlled in some instances (such as randomized assignment of program inputs to clients) or simply observed (such as the weather). The important thing is that the explanatory variable is not affected by the dependent variable. In economic applications, the explanatory variable is called a leading indicator for the dependent variable, and the model is called a leading-indicator model. In physical contexts, the model is called a transfer-function model.

We shall illustrate the development of a transfer-function model using an example presented in Box and Jenkins' book. The example involves sales and a leading indicator for sales. The data set is available from the Internet (there is a reference to this on CRAN website (the R Datasets Package, BJsales, "Sales Data with Leading Indicator", but it could not be located there).

Figure 13 plots the sales data and Figure 14 plots the leading indicator.

Figure 13. Sales Data

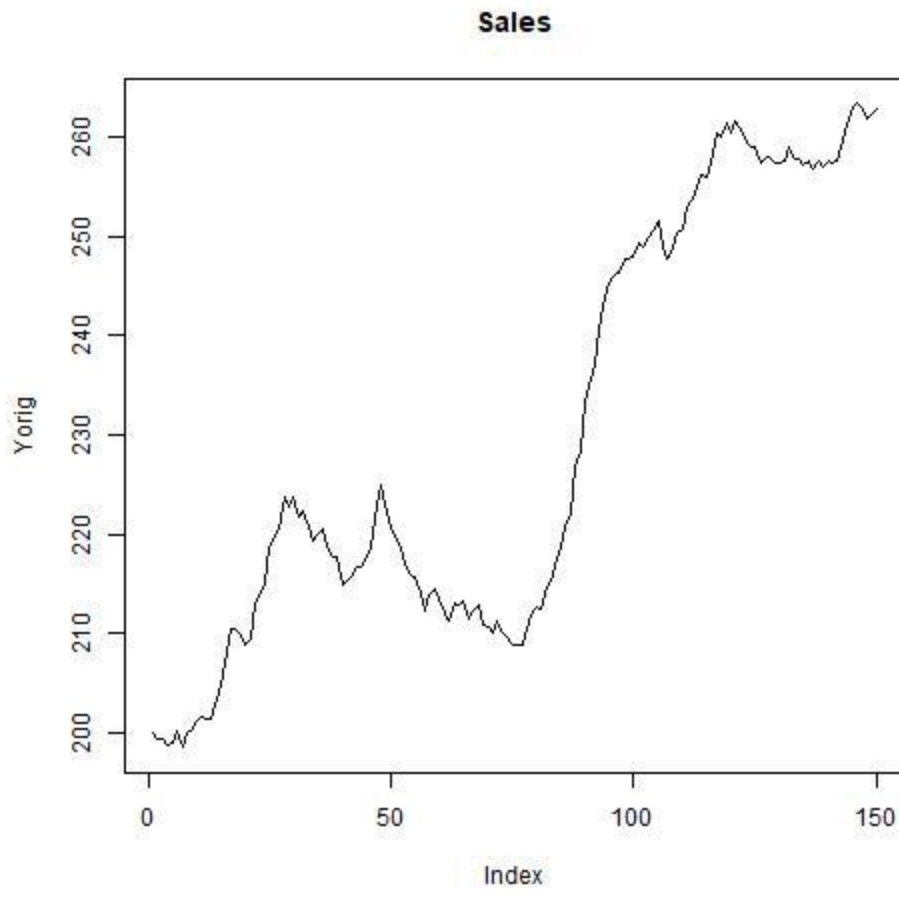
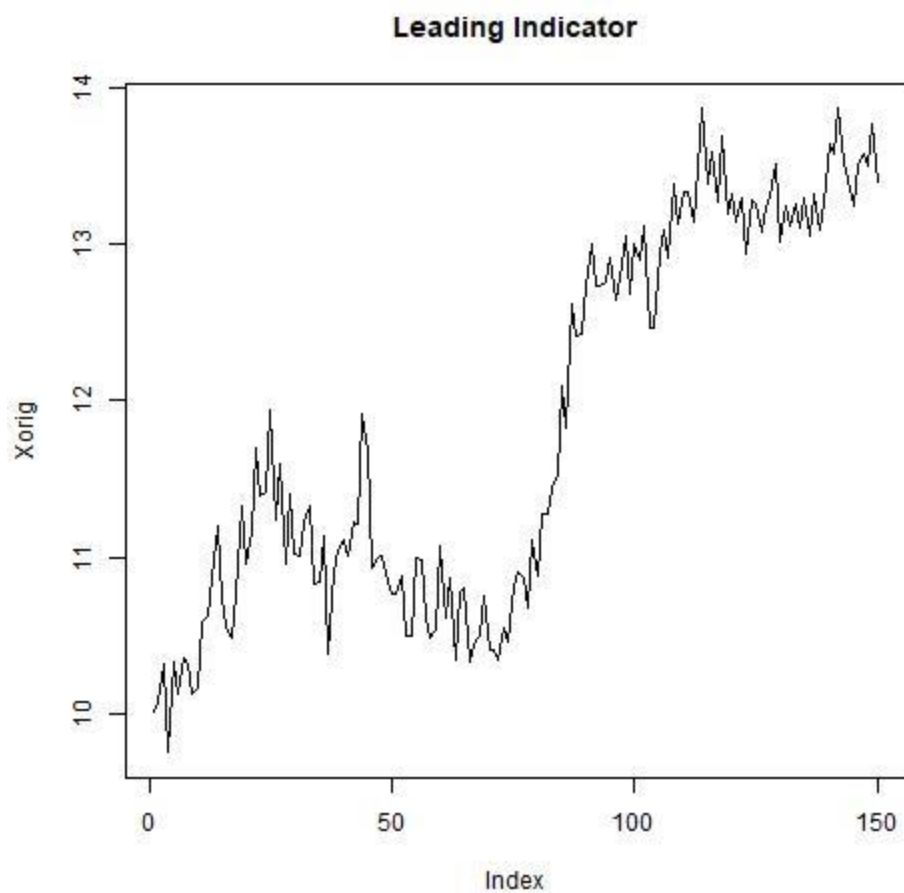


Figure 14. Leading Indicator Data



The sales and leading-indicator series are clearly nonstationary. They can be transformed to stationary variates by taking one-time-step differences. Figures 15 and 16 display plots of the first-differenced series.

Figure 15. Differenced Sales Data

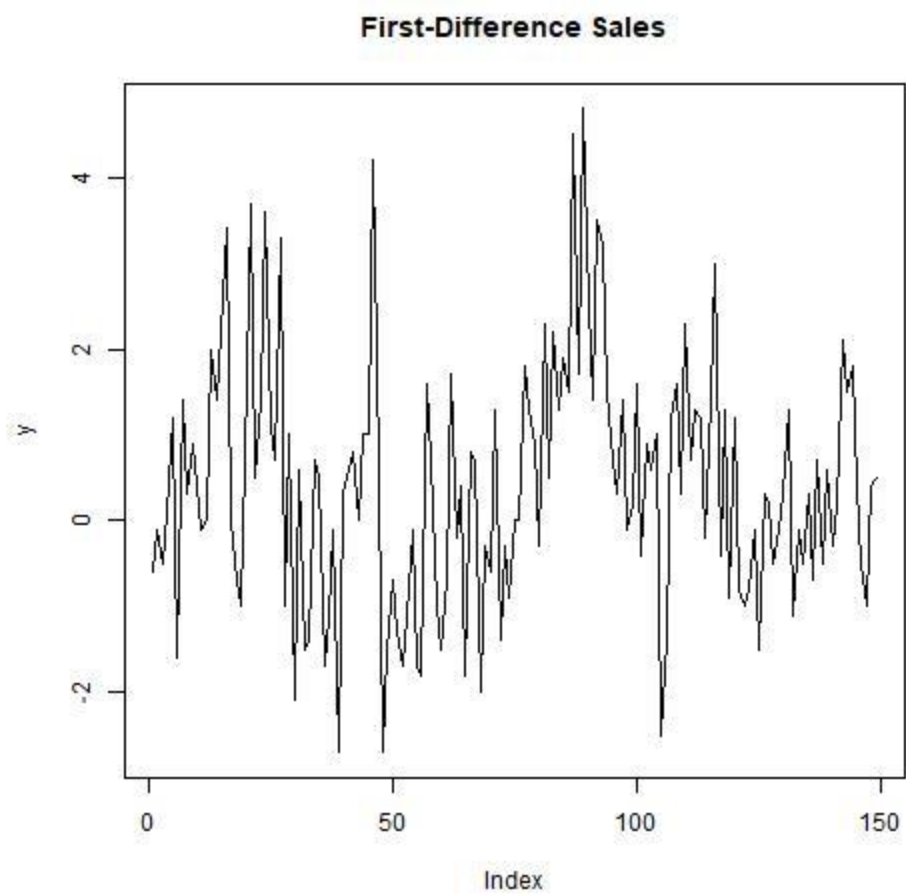
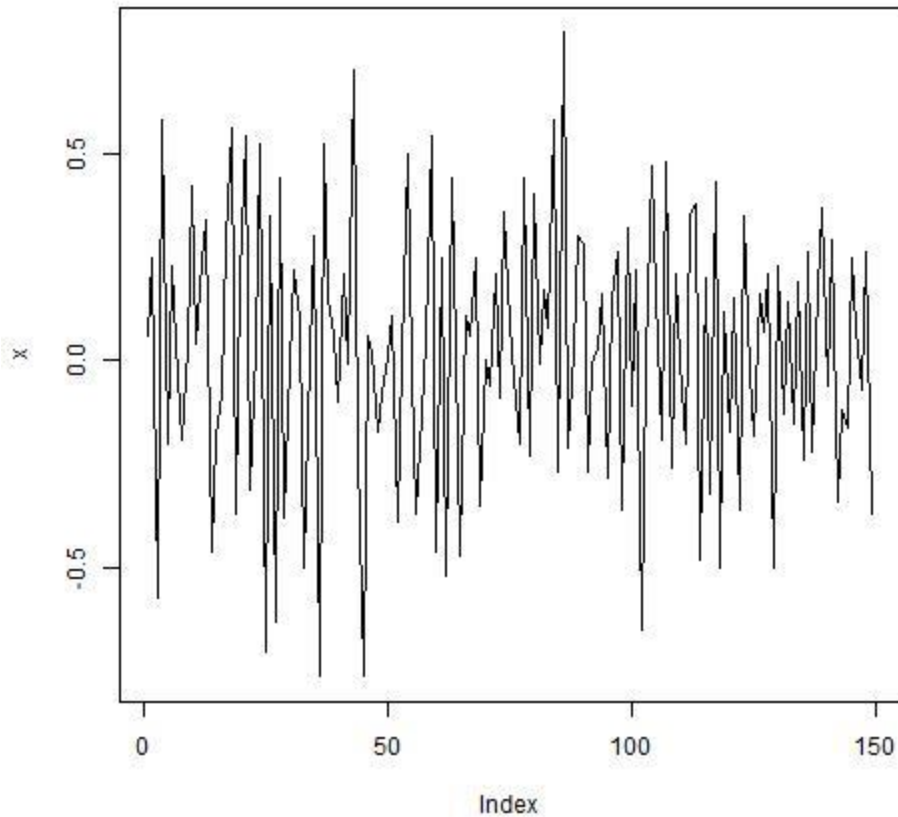


Figure 16. Differenced Leading Indicator

First-Difference Leading Indicator



The first-differenced series exhibit stationary behavior, and the model parameters will be estimated for a transfer function in these variates.

The formula for a transfer-function model involving a single input and a single output will now be discussed. In this model, it is assumed that the input and output processes are stationary. Hence, when developing this model using the sales/indicator data, the model development will be done in terms of the first-differenced variables, denoted as x (input) and y (output), where

$$x_t = (1 - B)x_{\text{orig},t} = x_{\text{orig},t} - x_{\text{orig},t-1}$$

$$y_t = (1 - B)y_{\text{orig},t} = y_{\text{orig},t} - y_{\text{orig},t-1}$$

where x_{orig} and y_{orig} denote the original (raw, untransformed) data.

The model to be estimated is

$$y_t = \delta^{-1}(B)\omega(B)x_{t-b} + n_t$$

where

$$y_t = \nabla^d Y_t$$

$$x_t = \nabla^d X_t$$

$$n_t = \nabla^d N_t$$

are stationary processes and

$$n_t = \phi^{-1}(B)\theta(B)a_t.$$

The notation here follows that of the 5th edition of the Box-Jenkins book, *Time Series Analysis, Forecasting and Control*.

Just as the autocorrelation function of a univariate univariable time series is useful for suggesting the structure of a univariate ARMA model, the cross-correlation function (CCF) of the input and output series is useful for suggesting the structure of a transfer-function model. In general, however, it is difficult to ascertain the model structure from the CCF of the input and output, even when these are stationary. After applying a “prewhitening” transformation to the two series, however, the CCF can be used very effectively to identify the model structure.

From this point forward, it we shall be working with the stationary variates x_t and y_t .

The prewhitening transformation is the transformation that transforms the input variable x_t into a white noise sequence. That is, if the ARMA model for x_t is

$$x_t = \phi_x^{-1} \theta_x \alpha_t,$$

then the transformation that transforms x_t into a white-noise series is

$$\alpha_t = \theta_x^{-1} \phi_x x_t.$$

It can be shown that the CCF between the prewhitened x_t series and the series obtained by applying the same transformation to the y_t is proportional to the impulse response function of the model.

This is easy to show. Let us write the transfer-function model in the form

$$y_t = v_0 x_t + v_1 x_{t-1} + v_2 x_{t-2} + \dots + n_t = v(B)x_t + n_t$$

where v_0, v_1, v_2 are the values of the impulse response function. Let us apply the prewhitening transformation $\theta_x^{-1} \phi_x$ to both sides of this model equation, to obtain:

$$\beta_t = \theta_x^{-1} \phi_x y_t = v(B) \theta_x^{-1} \phi_x x_t + \theta_x^{-1} \phi_x n_t = v(B) \alpha_t + \theta_x^{-1} \phi_x n_t$$

Now, multiply both sides of this equation by α_{t-k} and take expectations, to obtain:

$$\gamma_{\alpha\beta}(k) = v_k \sigma_\alpha^2$$

where

$$\gamma_{\alpha\beta}(k) = E[\alpha_{t-k}\beta_t]$$

Is the cross-covariance at lag +k between the series α_t and β_t .
Hence

$$v_k = \frac{\gamma_{\alpha\beta}(k)}{\sigma_\alpha^2}$$

or, in terms of cross-correlations,

$$v_k = \frac{\rho_{\alpha\beta}(k)\sigma_\beta}{\sigma_\alpha^2}, \quad k = 0,1,2, \dots$$

So, the cross-correlation function between the prewhitened input and the similarly transformed output is directly proportional to the impulse response function. This fact can be used to suggest a model structure, such as the size of the lag, b, between x and y, and the degrees of δ and ω .

The first step in the identification process is to obtain an ARMA representation for the input. Using the same identification process discussed earlier for the univariate univariable ARMA models, this is seen to be

$$x_t = \alpha_t - 0.4744\alpha_{t-1}$$

with $\sigma_\alpha^2 = .07794$.

The prewhitening filter is hence

$$(1 - .47B)^{-1}.$$

Applying this filter to the input yields a_t . The R commands for developing this model and this filter to the output (t) are

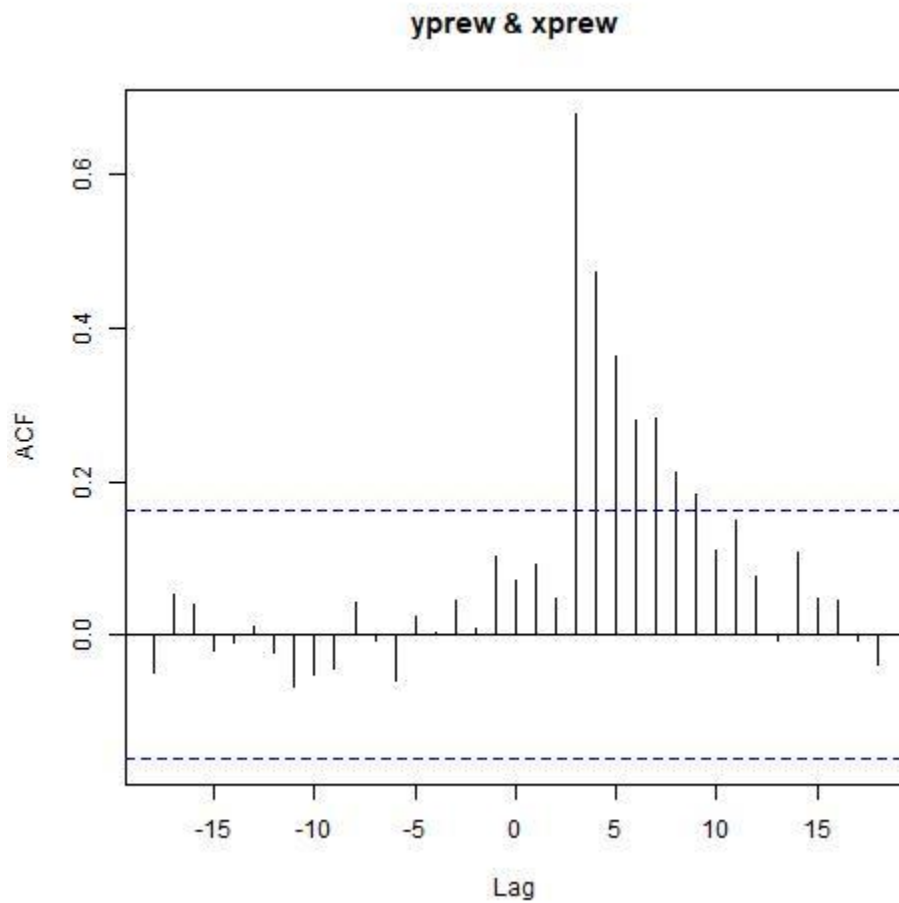
```
#Determine ARMA model for input, x.
nm1=arima(x,order=c(0,0,1))
print(nm1) # Prints the ARIMA coefficients for x
print(summary(nm1))
f1=c(-nm1$coef[1:1]) # Creates a filter to transform y
print(f1)
#Use convolution method for AR model, recursive method for MA
model.
#Note that this is different from the gas-furnace model.
yf = filter(y,f1,method=c("recursive"),sides=1)
print(yf)
yprew=yf[2:149] # transformed (i.e., prewhitened) y
print(yprew)
xprew=nm1$residuals[2:149] # transformed x
print(xprew)
#Check to see that filtering produces the same values of the
residuals.
xf = filter(x,f1,method=c("recursive"),sides=1)
print(xf)
```

The R commands for calculating the cross-correlation function of the prewhitened x and similarly transformed y are:

```
CCF =ccf(yprew,xprew, ylab="CCF", main="Cross-correlations
after prewhitening") # computes the cross-correlations
print(sd(yprew))
print(sd(xprew))
vk=(sd(yprew)/sd(xprew))*CCF$acf # impulse response function
print(vk)
```

Figure 17 shows a plot of the cross-correlation of the prewhitened x and similarly transformed y.

Figure 17. Cross-correlation function of prewhitened input and similarly transformed output



This plot clearly shows a lag of $b = 3$ between the input and output response, and that after accounting for a relationship between y and x , the resulting series exhibits autoregressive behavior.

This suggests a model of the form

$$y_t = \mu + \frac{\omega_0}{1 - \delta B} x_{t-3} + (1 - \theta B) a_t$$

The R code for estimating this model and examining the model residuals is

```
m1=tfm1(y,x,orderX=c(1,0,3),orderN=c(0,0,1))
print(m1)
print(summary(m1))
print(names(m1)) # check contents of output
print(m1$estimate) # model coefficients
print(m1$varcoef) # variances of model coefficients
print(m1$sigma2) # residual variance
print(acf(m1$residuals)) # acf of the residuals
print(ccf(m1$residuals,x, ylab="CCF")) # cross-correlation
between input series and residuals
print(ccf(m1$residuals,xprew, ylab="CCF")) # cross-correlation
between prewhitened input and residuals
```

The output for the model estimation is

Transfer function coefficients & s.e.:
in the order: constant, omega, and delta: 1 1 1

```
      [,1] [,2] [,3]
v    0.02811 4.6943 0.72646
se.v 0.00995 0.0558 0.00417
```

ARMA order:

```
[1] 0 0 1
```

ARMA coefficients & s.e.:

```
      [,1]
coef.arma 0.5247
se.arma   0.0697
```

```
> print(m1)
```

```
$estimate
```

```
[1] 0.02810906 4.69429153 0.72646203 0.52474147
```

```
$sigma2
```


[1] 0.04997414

That is, the estimated model parameters are $\mu = .02811$, $\omega = 4.6943$, $\delta = .7265$, $\sigma_a^2 = .04997$. The estimation command also estimates the ARMA model for the input, which is estimated to be

$$x_t = (1 - .5247B)\alpha_t$$

with $\sigma_\alpha^2 = .0697$.

The parameter estimates agree well with the results reported in the Box-Jenkins book (page 468), except for the parameter in the ARMA model for x , which is estimated as .32 rather than .4744 in the arima program and .5247 in the transfer-function program (with model error variance .0676 rather than .07794 (from the arima program) or .0697 (from the transfer-function estimation program)).

Figure 18 shows a plot of the ACF of the model residuals, Figure 19 shows a plot of the CCF between the model residuals and the input, and Figure 20 shows a plot of the CCF between the model residuals and the prewhitened input.

Figure 18. Plot of autocorrelation function of model residuals

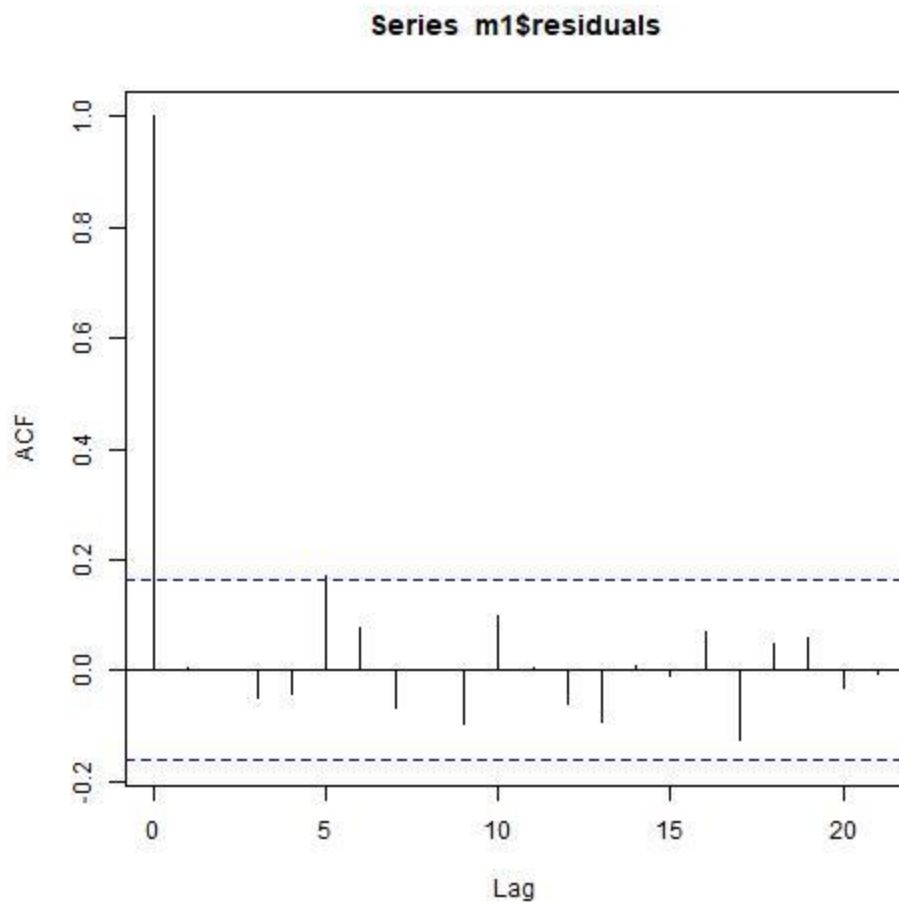


Figure 19. Plot of cross-correlation function of model residuals and input

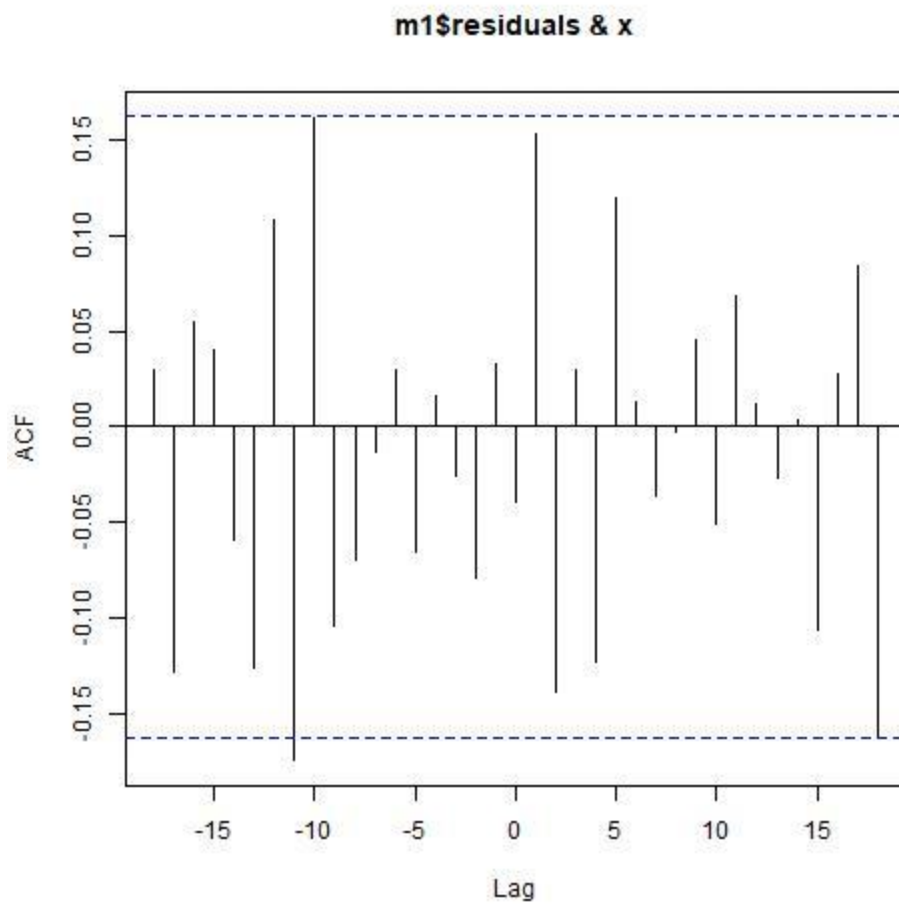
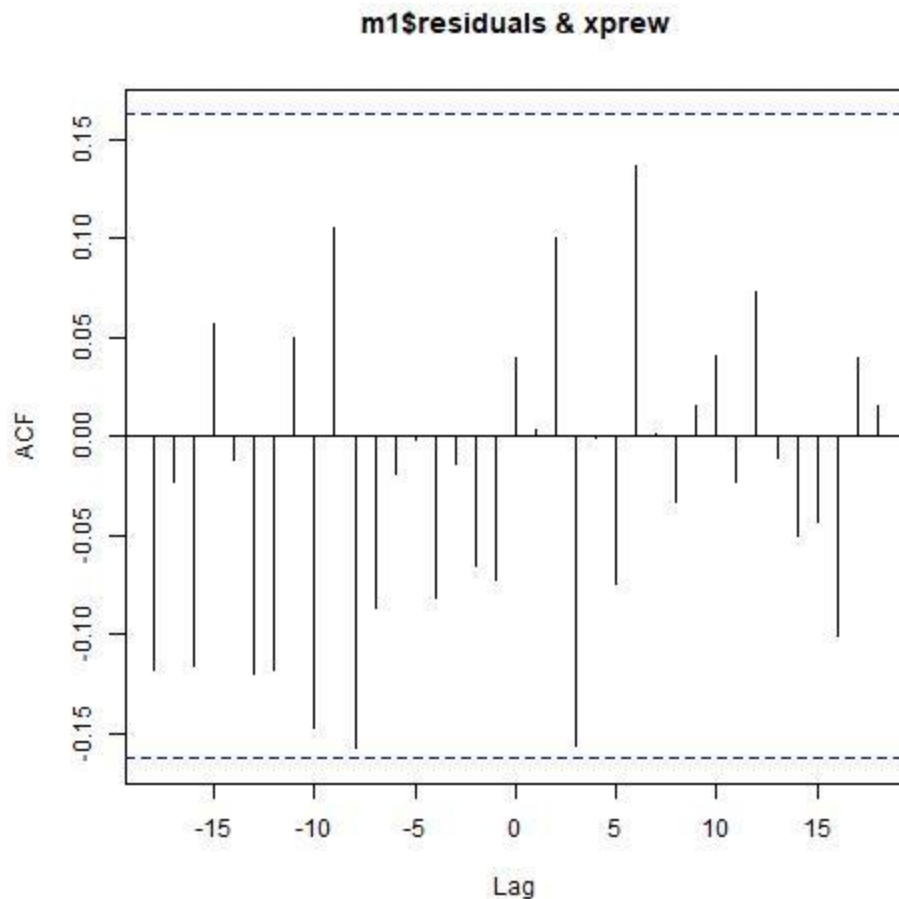


Figure 20. Plot of cross-correlation function of model residuals and prewhitened input.



These plots indicate that the estimated model is a good representation of the sales/leading indicator data.

Forecasts for the Transfer-Function Model

The R library, MTS, which contains the procedure for estimating transfer functions having one or two inputs, does not include a forecast procedure. To make predictions with this model in R, such an R procedure would have to be identified or developed. Other packages (than MTS) include ARMAX routines, but these were not investigated. The transfer-function model can be represented as a special case of the vector time series models to be considered in the next section, and software for analyzing the latter can be adapted to analysis of the transfer-function model.

A discussion of how to make forecasts for a transfer-function model using a multivariate model (which is discussed in the next section) is presented on page 512, of the Box-Jenkins book (5th edition), or on page 30 of the Tsay book.

Transfer-Function Models with More than One Input

The Box-Jenkins book discusses transfer function models having more than one input, and discusses the one-input and two-input cases in detail. A problem with models involving more than one input is that the number of parameters increases (roughly in proportion to the number of inputs), so that model development becomes more difficult. This is not a problem if the inputs are generated, as in a simulation model or designed experiment, but the task of model identification (specification of the structure of the model, prior to estimation) becomes difficult.

As the number of explanatory variables included in the model increases, the relative magnitude of the sources of variation shifts from the past to the present. With a large number of explanatory variables, the model essentially becomes a large multiple regression model, possibly with a serially correlated error term. This sort of model is generally referred to as a Cochrane-Orcutt or Prais-Winsten model. In these models, the model for the error term is restricted to an autoregressive model of order one. This restriction is not overly restricting, since as the number of explanatory variables included in a model, the serial correlation of the disturbances tends to decrease.

Models having a large number of explanatory variables are not very useful for forecasting. The problem is that to make forecasts, it is necessary to specify future values of the explanatory variables. In general, the future values of leading indicators are not known. The model can make reasonable

forecasts for a number of time steps into the future equal to the lag of the leading indicator(s). Models having a large number of explanatory variables are more useful for making estimates conditional on values of the explanatory variables in the current time period (such as estimates for different regions, at the present time), not estimates (predictions, forecasts) conditional on values of the explanatory variable in the future.

For models containing explanatory variables, there are two types of forecasts. The first type is unconditional forecasts. These forecasts may be calculated by forecasting the explanatory variables and using these forecasts for their future values. Note that for such forecasts, it does not matter whether the model of the relationship of the dependent variable to the explanatory variables is correctly specified. The second type of forecast is forecasts that are conditional on specified values of one or more input variables. This type of forecast is desired for policy analysis and control applications. For this type of forecast, it is necessary that the time-series model be developed from data in which the input variables were controlled in the same fashion as in the application (or from a causal model).

To reiterate, the essential restriction on model forecasts is that a model may be used to make forecasts only in a data-generating context similar to that for which the model was developed. If the model is developed from input data that were controlled, then the model may be used to forecast output conditional on similar changes to the control variables. If the model is developed from input data that were simply observed (not controlled), then, absent a valid causal model that describes the causal relationships among the model variables, the model is not useful for making predictions when the input is controlled.

Summary

This example demonstrates the power of the Box-Jenkins methodology to represent complex multidimensional processes very well and efficiently (i.e., with a very small number of parameters).

The amount of time involved to perform the data processing, analysis, estimation, and documentation for this model development, for someone familiar with the Box-Jenkins methodology and the R commands involved to implement it, is on the order of a couple of hours (for a “clean” data set).

Multivariate models (VAR, VARMA and VARMAX models)

For the univariate models just discussed, the model could actually involve more than one random variable – the error term for the model of the dependent variable conditional on the independent variables, and the error terms for the model(s) of the independent variables. The essential requirement was that the error term for the model of the dependent variable conditional on the independent variables was a one-dimensional random variable. The situation may be described by saying that there is a univariate response variable, or a response variable having a one-dimensional probability distribution function (conditional on the explanatory variables).

We shall now consider the case in which there are more than one response variables, and they have a nontrivial multidimensional joint distribution function. A multivariate random variable is represented as a vector of random variables having a nondegenerate multidimensional joint probability distribution. Examples of multivariate variables include:

Prices of currency pairs in different markets, such as EUR/USD, GBP/USD and USD/CHF.

US unemployment rate, inflation rate, and interest rate.

Price and volume data for a financial instrument (such as a stock or commodity price)

For the examples presented above, the components of the multivariate random variable mutually affect each other. (They could be stochastically independent, but in that case the behavior of the individual components could be analyzed using univariate models.)

The time series models used to describe the behavior of multivariate random variables are a direct extension of the models used to describe the behavior of univariate random variables. The model formulas are similar, with some scalars replaced by vectors and some vectors replaced by matrices.

In the univariate case, moving average components are common. Moving average components are less common in multivariate time series models, because when several variables are included in a model, the autocorrelational complexity of the model error term often decreases.

Univariate time series models are called AR, MA, ARMA, ARIMA and ARMAX. The terminology is analogous for multivariate time series models, with the prefix term “vector” added, as in VAR, VMA, VARMA, VARMAX. The term “integrated” is usually not included in the vector acronyms.

Let $\mathbf{Z}_t = (z_{1t}, \dots, z_{kt})'$, $t = 0, \pm 1, \pm 2, \dots$, denote a k -dimensional time-series vector of random variables. The following is a general formula for a vector autoregressive moving average (or VARMA) process.

$$(\mathbf{Z}_t - \boldsymbol{\mu}) = \sum_{j=1}^p \boldsymbol{\Phi}_j (\mathbf{Z}_{t-j} - \boldsymbol{\mu}) = \mathbf{a}_t - \sum_{j=1}^q \boldsymbol{\Theta}_j \mathbf{a}_{t-j}$$

where \mathbf{a}_t is a vector white noise process with mean vector $\mathbf{0}$ and covariance matrix $\Sigma = E[\mathbf{a}_t \mathbf{a}_t']$. The process is referred to as a VARMA process whether \mathbf{Z}_t is stationary or not.

A vector white-noise process is a sequence of random vectors $\dots, \mathbf{a}_1, \dots, \mathbf{a}_t, \dots$ with $\mathbf{a}_t = (a_{1t}, \dots, a_{kt})'$ such that $E[\mathbf{a}_t] = \mathbf{0}$, $E[\mathbf{a}_t \mathbf{a}_t'] = \Sigma$, and $E[\mathbf{a}_t \mathbf{a}_{t+r}'] = \mathbf{0}$ for $r \neq 0$. Its covariance matrices $\Gamma(r)$ are given by

$$\Gamma(r) = E[\mathbf{a}_t \mathbf{a}_{t+r}'] = \begin{cases} \Sigma & \text{for } r = 0 \\ \mathbf{0} & \text{for } r \neq 0 \end{cases}$$

The $k \times k$ covariance matrix Σ is assumed to be positive definite (otherwise the dimension k of the process could be reduced).

While many features of vector time series models are similar to those for univariate time series models, there are some significant differences:

A multivariate process with model as represented above is stationary if the roots of $\det\{\mathbf{I} - \Phi B\} = 0$ exceed one in absolute value.

A multivariate process is invertible if the roots of $\det\{\Theta(B)\}$ exceed one in absolute value.

The parametric representation of a VARMA process is not unique, that is, different parametric representations may give rise to the same infinite MA representation. (VAR and MA models are unique.) This means that the VARMA model is not identified. This problem is overcome by imposing restrictions on the structural specification.

As the number of variables (dimensionality) of the process increases, the number of model parameters increases rapidly. This situation leads to difficulties in identification, estimation, interpretation, validation and application.

Additional methodologies are available to assist identification of the model structure.

The phenomenon of cointegration.

Detailed explanations of vector autoregressive moving average processes, including sample R code to conduct analysis, are presented in a number of texts, including the books by Box, Jenkins, Reinsel and Lyung; Tsay, and Lütkepohl (cited in the References section at the end).

Rather than reproduce an example from one of these texts, we shall present an example based on an article about vector autoregressive processes that is available for free on the Internet:

Stock, James H. and Mark W. Watson, “Vector Autoregressions,” *Journal of Economic Perspectives*—Volume 15, Number 4 —Fall 2001—Pages 101–115, posted at <https://pubs.aeaweb.org/doi/pdfplus/10.1257/jep.15.4.101>

We shall perform the analysis described in this article using procedures available in the MTS R package (which is used in the Box-Jenkins and Tsay books).

The approach used for identification of the structure of multivariate time series differs somewhat from that used for univariate multivariable models. A reason for this is that in many multivariate applications, it is desired to estimate impulse responses for a larger number of time-steps into the future than in many short-term-forecasting applications. To do this, it is necessary to develop a model that retains information about the levels of the variables, not just fluctuations in them. For this reason, it is not desirable to difference the data, or to prewhiten it, since those transformations removes the long-term mean and the local level. Another reason for not applying differencing to variates prior to model development in multivariate applications is that different series may have different orders of integration (i.e., the number of differences required to achieve stationarity).

Applying the same number of differences to all series when one series did not require it would lead to a noninvertible model. Another situation in which uniform differencing is inappropriate is in the analysis of cointegrated series (for which converting each series to a stationary series prior to the analysis would lose information about cointegration).

A problem that arises is that in the absence of prewhitening, it is difficult to use the cross-correlation function as a guide to model structure. For multivariate time series analysis, a more useful guide to identifying model structure are the various “information” criteria, such as the Akaike Information Criterion (AIC), the Bayes Information Criterion (BIC) and the Hannan-Quinn Information Criterion (HQC).

Figure 21 shows a plot of the three series of interest in the Stock-Watson article. They are quarterly inflation rate, unemployment rate, and interest rate (federal funds rate) for the United States. These data are available from the Federal Reserve Bank of St. Louis website (Federal Reserve Economic Data (“FRED”) at Internet website <https://fred.stlouisfed.org/>). For the Stock-Watson article, data were used from the time period 1960-2000. For the analysis presented here, data are used for the time period 1970 – 2019. Since the analysis time period is somewhat different, the analysis results differ somewhat.

Figure 21a. Monthly US Inflation Rate, 1970-2019

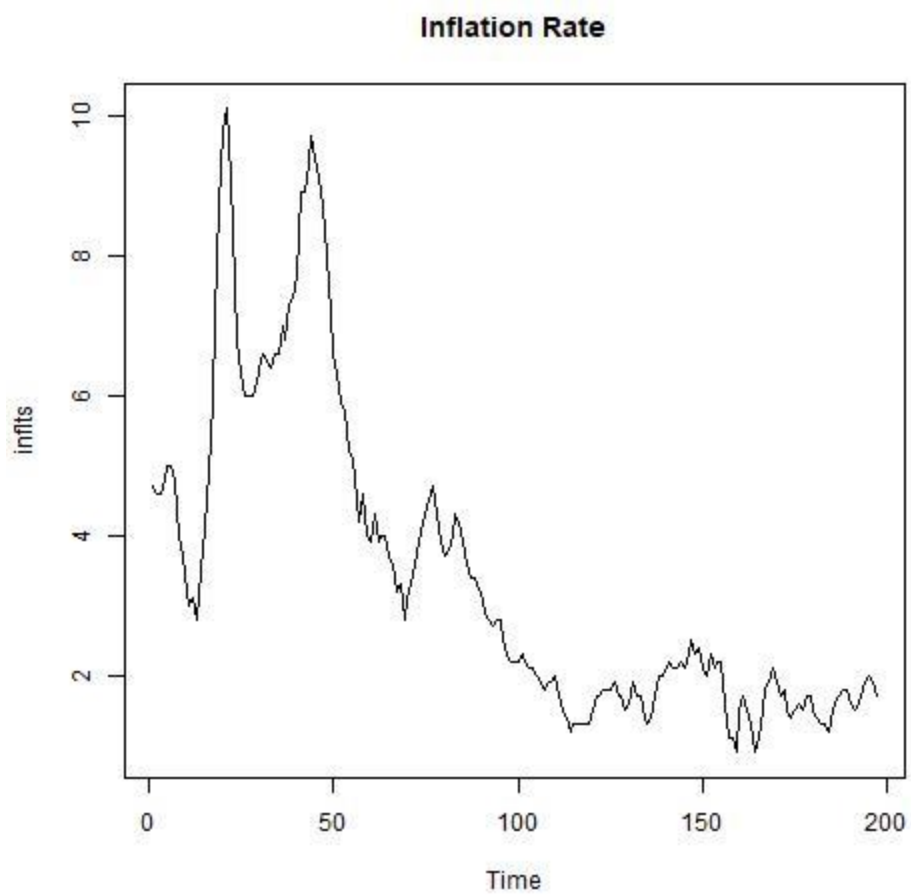


Figure 21b. Monthly US Unemployment Rate, 1970-2019

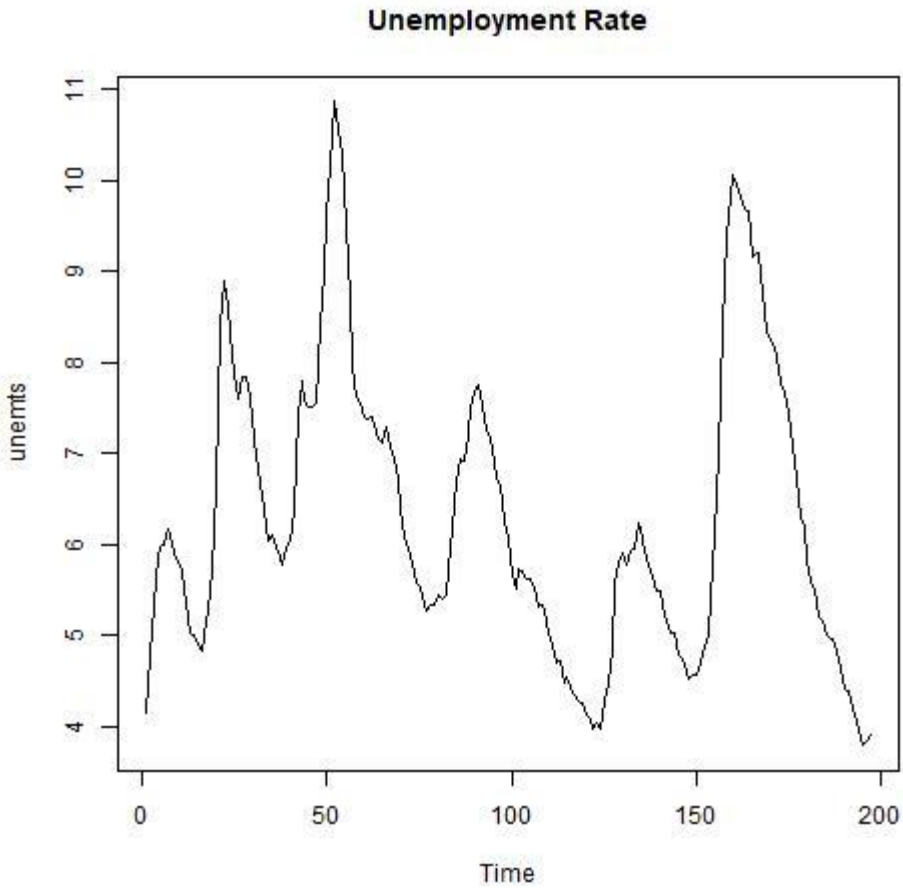


Figure 21c. Monthly US Interest Rate, 1970-2019

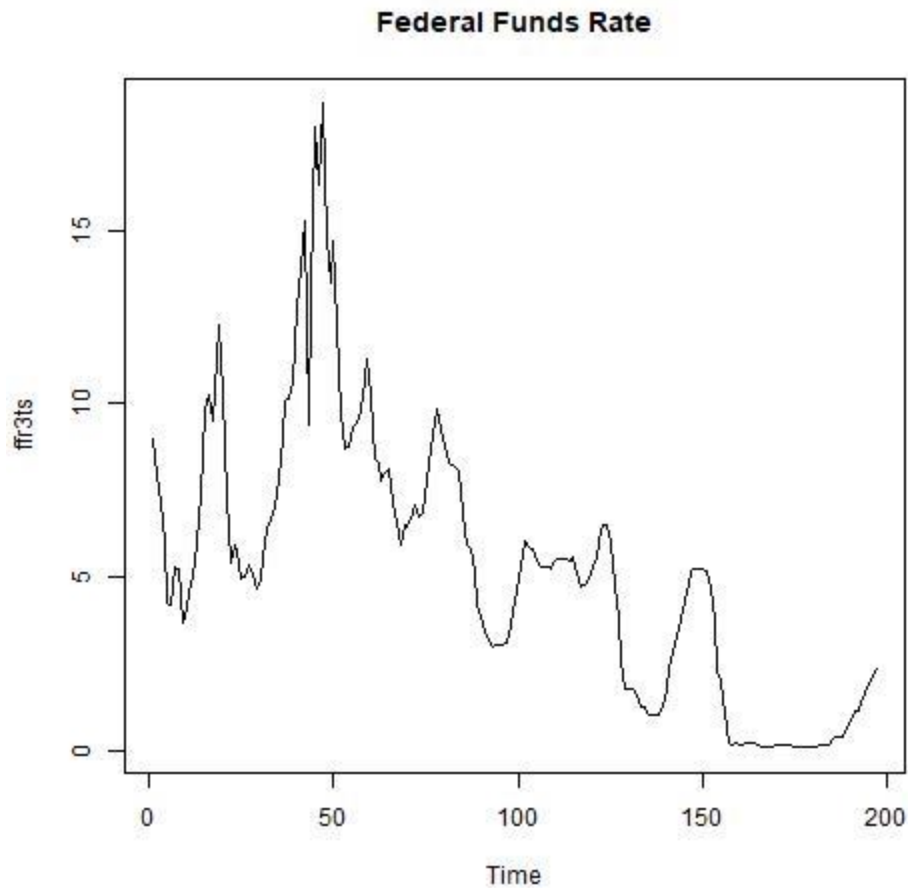


Figure 21 shows a plot of the three information criteria. The lowest curve is the AIC, the middle curve the BIC, and the upper curve the HQC. This plot suggests that a VAR of order 3 or four should be reasonable. Since the Stock-Watson article used order 4, that is used here.

The MTS R command for fitting a VAR of order 4, and the model output from program VARMA, is as follows.

```
m2=VARMA(x, p = 4, q = 0, include.mean = T, fixed = NULL,
beta=NULL, sebeta=NULL, prelim = F, details = F, thres = 2)
```

With multivariate models, there are a very large number of parameters involved (both model coefficients and model error

covariances). For the model specified above, there are 39 parameters. The parameters are correlated, and their interpretation is difficult. It is more customary to present results for a multivariate VAR or VARMA model in terms of the forward error variance decomposition and the impulse response function.

The MTS R command for the forward error variance decomposition is

```
mFEV=FEVdec(Phi = m2$Phi, Theta = m2$Theta, Sig =
m2$Sigma, lag = 12)
```

and the corresponding model output is

Order of the ARMA model:

```
[1] 4 0
```

Standard deviation of forecast error:

```
      [,1] [,2] [,3] [,4] [,5] [,6] [,7]
[1,] 0.2824465 0.4948000 0.7153921 0.9130447 1.0851310
1.231866 1.359084
[2,] 0.2290792 0.4284132 0.6155420 0.7810687 0.9220169
1.038173 1.131666
[3,] 0.8303473 1.2903207 1.5572701 1.8140999 2.0486092
2.242632 2.411987
      [,8] [,9] [,10] [,11] [,12] [,13]
[1,] 1.469349 1.565370 1.649479 1.723358 1.788350 1.845651
[2,] 1.205451 1.262725 1.306838 1.340906 1.367588 1.389067
[3,] 2.564053 2.696413 2.812222 2.915143 3.006609 3.088375
```

Forecast-Error-Variance Decomposition

Forecast horizon: 1

```
      [,1] [,2] [,3]
[1,] 1.000000000 0.000000000 0.000000000
[2,] 0.005533371 0.99446663 0.000000000
[3,] 0.007649980 0.05437214 0.9379779
```

Forecast horizon: 2

[,1] [,2] [,3]
[1,] 0.990185337 0.005002979 0.004811684
[2,] 0.003643369 0.991302563 0.005054068
[3,] 0.070694754 0.201267387 0.728037859
Forecast horizon: 3

[,1] [,2] [,3]
[1,] 0.96086409 0.01743783 0.02169808
[2,] 0.01313491 0.96984853 0.01701656
[3,] 0.06450454 0.31051627 0.62497919
Forecast horizon: 4

[,1] [,2] [,3]
[1,] 0.92924655 0.0318501 0.03890335
[2,] 0.02261812 0.9408154 0.03656650
[3,] 0.06325783 0.3726609 0.56408127
Forecast horizon: 5

[,1] [,2] [,3]
[1,] 0.90393099 0.04650739 0.04956162
[2,] 0.03147566 0.90851331 0.06001103
[3,] 0.07111132 0.42251055 0.50637813
Forecast horizon: 6

[,1] [,2] [,3]
[1,] 0.88406724 0.05966061 0.05627215
[2,] 0.03894620 0.87716323 0.08389056
[3,] 0.07952439 0.46351539 0.45696022
Forecast horizon: 7

[,1] [,2] [,3]
[1,] 0.86855360 0.0714132 0.0600332
[2,] 0.04557927 0.8472834 0.1071373
[3,] 0.08896826 0.4905841 0.4204477
Forecast horizon: 8

[,1] [,2] [,3]
[1,] 0.85659892 0.0817998 0.06160128
[2,] 0.05185917 0.8188446 0.12929622
[3,] 0.10099468 0.5080493 0.39095604
Forecast horizon: 9


```

      [,1] [,2] [,3]
[1,] 0.84728488 0.09079898 0.06191614
[2,] 0.05805979 0.79197325 0.14996696
[3,] 0.11391896 0.51920230 0.36687873

```

Forecast horizon: 10

```

      [,1] [,2] [,3]
[1,] 0.8399831 0.09848148 0.06153537
[2,] 0.0642759 0.76676428 0.16895982
[3,] 0.1274378 0.52475940 0.34780276

```

Forecast horizon: 11

```

      [,1] [,2] [,3]
[1,] 0.83430054 0.1049582 0.06074122
[2,] 0.07056065 0.7433304 0.18610897
[3,] 0.14172129 0.5259661 0.33231261

```

Forecast horizon: 12

```

      [,1] [,2] [,3]
[1,] 0.8299265 0.1103278 0.05974564
[2,] 0.0769084 0.7218599 0.20123166
[3,] 0.1563640 0.5241587 0.31947725

```

Forecast horizon: 13

```

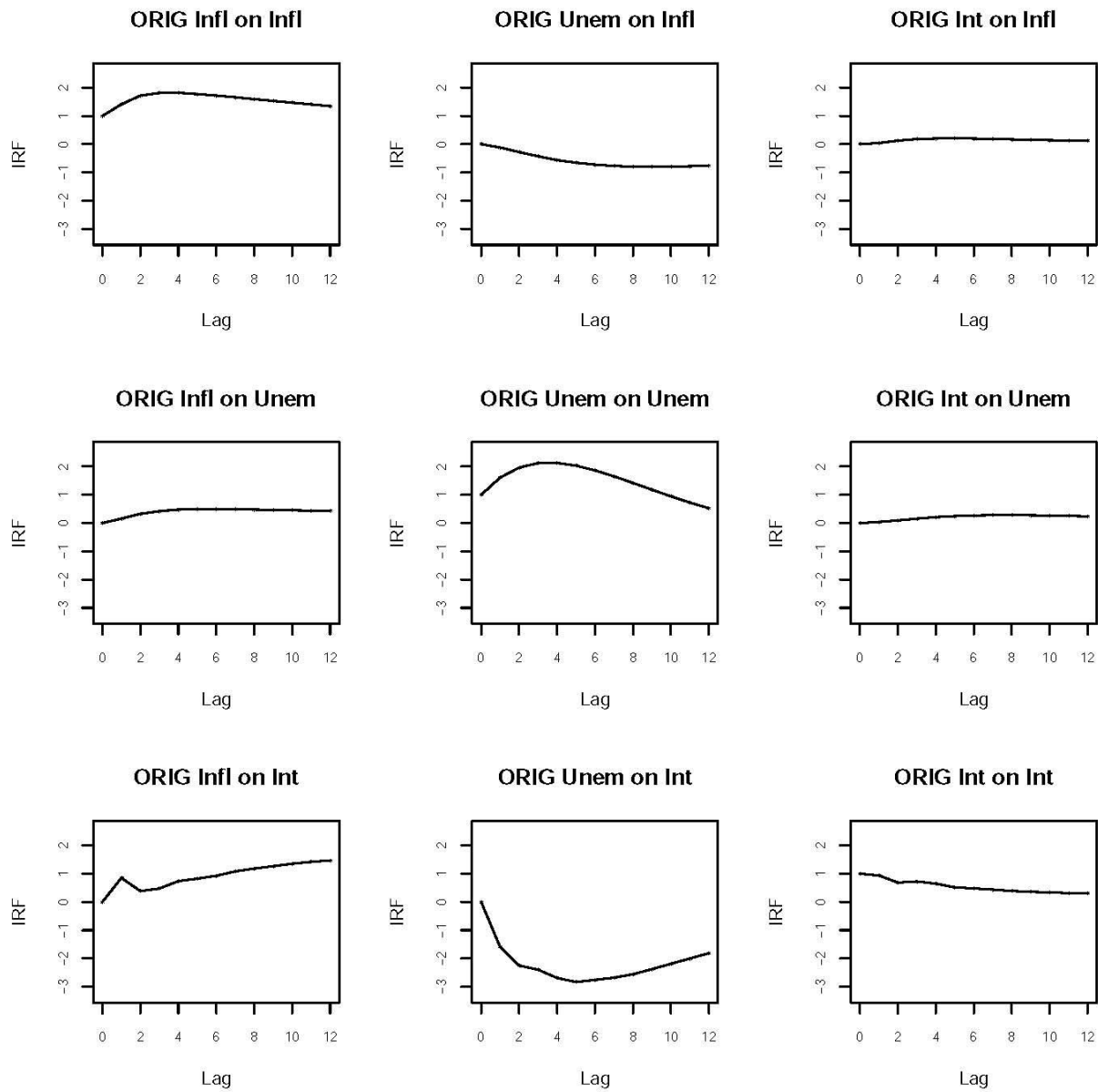
      [,1] [,2] [,3]
[1,] 0.82661386 0.1147072 0.05867889
[2,] 0.08328338 0.7025004 0.21421627
[3,] 0.17107228 0.5201138 0.30881392

```

The MTS R command to generate the impulse response function is:

```
mirf1=VARMAirf(Phi = m2$Phi, Theta = m2$Theta, Sigma = m2$Sigma, lag = 12, orth = FALSE)
```

The graph of the impulse response function is presented in Figure 21.



These results are in general agreement with those presented in the Stock-Watson article. Note that the impulse response function graphs presented in the article are transposed from those presented here.

Because of the correlations among the variates, changes in one of them will introduce changes in the others. To address this issue, it is standard practice to transform the original variables to

orthogonal variables (principal components, singular value decomposition, orthogonalization), and present the impulse response functions in terms of the transformed variates. The MTS R code calculates impulse response functions for both the original variates (“ORIG”) and the orthogonalized variates (“ORTH”), but graphs are included here only for the original variates.

The MTS R code to make forecasts is VARMApred(m2,12), and the output produced by this command is

```
> VARMApred(m2,12)
Predictions at origin 197
      infl unem ffr3
[1,] 1.630 3.962 2.391
[2,] 1.639 4.043 2.521
[3,] 1.701 4.159 2.566
[4,] 1.779 4.292 2.549
[5,] 1.862 4.431 2.543
[6,] 1.939 4.567 2.541
[7,] 2.011 4.696 2.534
[8,] 2.077 4.814 2.537
[9,] 2.139 4.921 2.549
[10,] 2.197 5.017 2.569
[11,] 2.251 5.101 2.597
[12,] 2.301 5.175 2.634
Standard errors of predictions
      [,1] [,2] [,3]
[1,] 0.2824 0.2291 0.8303
[2,] 0.4948 0.4284 1.2903
[3,] 0.7154 0.6155 1.5573
[4,] 0.9130 0.7811 1.8141
[5,] 1.0851 0.9220 2.0486
[6,] 1.2319 1.0382 2.2426
[7,] 1.3591 1.1317 2.4120
```

[8,] 1.4693 1.2055 2.5641
[9,] 1.5654 1.2627 2.6964
[10,] 1.6495 1.3068 2.8122
[11,] 1.7234 1.3409 2.9151
[12,] 1.7883 1.3676 3.0066

Summary

This section shows that very powerful R software is available for implementing VAR models. Similar software is available for MA and VARMA models, but only the VAR models are illustrated here. As mentioned earlier, once several variables are included in a model, the need to include AR or MA components is lessened. MA terms may be included in a multivariate time series model for a number of reasons, including:

An MA or VARMA representation may be a more efficient representation (i.e., involve fewer parameters or achieve a higher level of precision for a similar number of parameters) of a time series than a VAR representation. Overdifferencing may incorporate moving-average behavior. The nature of a physical process (such as the replacement of sample units in a rotation panel sample) may incorporate moving-average behavior.

Special Topics (ARCH and GARCH Models; Kalman Filter; Bayesian Estimation; Structural Models; Econometric Models)

ARCH and GARCH Models

For the models discussed thus far, the assumption has been made that the variance (or covariance) of the model error term is constant, or “homoscedastic.” In many applications, this assumption does not hold. In financial series, for example, it is

common for the variance of a process to increase when trading activity increases. A class of models has been developed that allow for the process variance to vary. These models are called “conditionally heteroscedastic,” since the variance is constant given certain past information.

The basic process of this type is the autoregressive conditional heteroscedasticity model, or ARCH model. In that model, the process error variance is assumed to depend on past values of the squared model error term. The basic model may be a regression-type model or an ARMA model.

The basic ARCH model may be formulated as follows. Suppose that a_t denotes the model error term, such as in an ARMA model or a regression model. It is assumed that

$$a_t = \sigma_t e_t$$

where e_t is a sequence of random variables with mean 0 and variance 1, and

$$\sigma_t^2 = \gamma_0 + \gamma_1 a_{t-1}^2 + \cdots + \gamma_m a_{t-m}^2$$

where $\gamma_0 > 0$, $\gamma_i \geq 0$ for $i = 1, \dots, m-1$, and $\gamma_m > 0$. That is, the model error variance σ_t^2 follows a moving average process. The inequality constraints assure that σ_t^2 is positive. The additional constraint

$$\sum_{i=1}^m \gamma_i < 1$$

ensures that the a_t are covariance stationary with finite unconditional variance σ_a^2 .

The preceding model is called an ARCH model of order m , or ARCH(m).

In the preceding formulation, the model for the error variance is a moving average model. (The adjective “autoregressive” in ARCH refers to the main process, not to the variance process.) The preceding ARCH model may be extended to a generalized autoregressive conditional heteroscedasticity model, or GARCH model, by allowing the process for the model error variance to be an ARMA process:

$$\sigma_t^2 = \gamma_0 + \gamma_1 a_{t-1}^2 + \cdots + \gamma_m a_{t-m}^2 + \delta_1 \sigma_{t-1}^2 + \delta_2 \sigma_{t-2}^2 + \cdots + \delta_k \sigma_{t-k}^2.$$

This model is called a GARCH model of order (m,k) or GARCH(m,k).

An example of an ARCH regression-type model is:

$$y_t = \mathbf{X}_t \boldsymbol{\beta} + a_t$$

where the variance, σ_t^2 , of a_t follows a MA process.

There are a variety of ARCH-type models. For example, the volatility of a financial time series often increases when the series is falling. A model that represents this type of behavior is the nonlinear asymmetric GARCH, or NAGARCH.

Routines for estimating ARCH-type models are available in a number of R libraries, for example procedure `garch` in the `tseries` package (for univariate time series) or the `gogarch` routine in the Generalized Orthogonal GARCH (GO-GARCH) model package. Examples will not be presented here.

Kalman Filtering

The models and R software discussed above assumed a high degree of regularity in both the models and the data. In general, it was assumed that data were available at regularly spaced time intervals, that there were no missing data, and that the model parameters were time-invariant.

A more general class of models, that relaxes these assumptions, is the class of Kalman filter models. The Kalman filter was developed by Richard Kalman in 1960 as a general recursive algorithm for predicting the state of a dynamic system. It is based on the following state-space representation of a system. (The notation here follows James D. Hamilton, *Time Series Analysis* (Wiley, 1994), pp. 372-408.)

For the discussion here, we shall consider the case in which the model parameters are not time-varying.

Let \mathbf{y}_t denote an $(n \times 1)$ vector of variates observed at time t . Let the $(r \times 1)$ possibly unobserved vector ξ_t denote the state (or state vector) of the system at time t . The state-space representation of the dynamics of \mathbf{y}_t is given by the following system of equations:

$$\begin{aligned}\xi_{t+1} &= F\xi_t + \mathbf{v}_{t+1} \\ \mathbf{y}_t &= A'\mathbf{x}_t + H'\xi_t + \mathbf{w}_t\end{aligned}$$

where F , A' and H' are matrices of parameters of dimension $(r \times r)$, $(n \times k)$, and $(n \times r)$, respectively, and \mathbf{x}_t is a $(k \times 1)$ vector of exogenous or predetermined variables. The first equation is called the state equation (or the transition equation, or the plant equation), and the second equation is called the observation equation (or the measurement equation). The $(r \times 1)$ vector \mathbf{v}_t , and the $(n \times 1)$ vector \mathbf{w}_t are vector white noise processes:

$$E(\mathbf{v}_t \mathbf{v}_t') = \begin{cases} Q & \text{for } t = \tau \\ \mathbf{0} & \text{otherwise} \end{cases}$$

$$E(\mathbf{w}_t \mathbf{w}_t') = \begin{cases} R & \text{for } t = \tau \\ \mathbf{0} & \text{otherwise} \end{cases}$$

Where Q and R are (r x r) and (n x n) matrices, respectively. The disturbances \mathbf{v}_t and \mathbf{w}_t are assumed to be uncorrelated at all lags:

$$E(\mathbf{v}_t \mathbf{w}_t') = \mathbf{0} \text{ for all } t \text{ and } \tau.$$

The statement that x_t is exogenous or predetermined means that x_t provides no information about ξ_{t+s} or \mathbf{w}_{t+s} for $s = 0, 1, 2, \dots$ beyond that contained in $y_{t-1}, y_{t-2}, \dots, y_1$. For example, x_t could include lagged values of y or variables that are uncorrelated with ξ_τ and \mathbf{w}_τ for all τ .

There are a number of R packages that provide routines for Kalman filtering. A review of some of them is provided in the article “Kalman Filtering in R,” by Fernando Tusell (*Journal of Statistical Software*, March 2011, Volume 39, Issue 2. <http://www.jstatsoft.org/>). The Kalman Filter and Smoother (KFAS) package, described at <https://cran.r-project.org/web/packages/KFAS/index.html>, implements most of the algorithms described in the book, *Time Series Analysis by State Space Methods* by J. Durbin and S. J. Koopman 2nd ed. (Oxford University Press, 2012).

One of the interesting features of the Kalman filter is that the forecast function is the same for Bayesian and nonBayesian implementations.

Bayesian Estimation

There are a large number of packages available in R to perform Bayesian estimation. A recent summary of such article is presented in the article “CRAN Task View: Bayesian Inference” by

Jong Hee Park, posted at <https://cran.r-project.org/web/views/Bayesian.html> .

Bayesian models are of particular importance for applications in developing countries, in which the amount of time series data required to develop time series models is limited or inadequate.

Structural Models

Forecasting models may be classified into two major categories, structural models and non-structural models. Here follows a description of structural models from the Wikipedia article on structural estimation.

“Structural estimation is a technique for estimating deep "structural" parameters of theoretical economic models. The term is inherited from the simultaneous equations model. In this sense "structural estimation" is contrasted with "reduced-form estimation," which is the statistical relationship between observed variables.

The difference between a structural parameter and a reduced-form parameter was formalized in the work of the Cowles Foundation. A structural parameter is also said to be "policy invariant" whereas the value of reduced-form parameter can depend on exogenously determined parameters set by public policy makers. The distinction between structural and reduced-form estimation within "microeconomics" is related to the Lucas critique of reduced-form macroeconomic policy predictions.

The original distinction between structure and reduced-form was between the underlying system and the direct relationship between observables implied by the system. Different combinations of structural parameters can imply the

same reduced-form parameters, so structural estimation must go beyond the direct relationship between variables.

Many economists now use the term "reduced form" to mean statistical estimation without reference to a specific economic model. For example, a regression is often called a reduced-form equation even when no standard economic model would generate it as the reduced form relationship between variables.

These conflicting distinctions between structural and reduced form estimation arose from the increasing complexity of economic theory since the formalization of simultaneous equations estimation. A structural model often involves sequential decisions making under uncertainty or strategic environments where beliefs about other agents' actions matter. Parameters of such models are estimated not with regression analysis but non-linear techniques such as generalized method of moments, maximum likelihood, and indirect inference. The reduced-form of such models may result in a regression relationship but often only for special or trivial cases of the structural parameters.”

Related to the issue of structural equation modeling is the Lucas critique, that discusses the difficulty in using large-scale econometric models for policy analysis. Here follows discussion from the Wikipedia article on that topic.

The Lucas critique, named for Robert Lucas's work on macroeconomic policymaking, argues that it is naive to try to predict the effects of a change in economic policy entirely on the basis of relationships observed in historical data, especially highly aggregated historical data. More formally, it states that the decision rules of Keynesian models—such as the consumption function—cannot be considered as

structural in the sense of being invariant with respect to changes in government policy variables. The Lucas critique is significant in the history of economic thought as a representative of the paradigm shift that occurred in macroeconomic theory in the 1970s towards attempts at establishing micro-foundations.

The basic idea pre-dates Lucas's contribution—related ideas are expressed as Campbell's law and Goodhart's law—but in a 1976 paper, Lucas drove to the point that this simple notion invalidated policy advice based on conclusions drawn from large-scale macroeconomic models. Because the parameters of those models were not structural, i.e. not policy-invariant, they would necessarily change whenever policy (the rules of the game) was changed. Policy conclusions based on those models would therefore potentially be misleading. This argument called into question the prevailing large-scale econometric models that lacked foundations in dynamic economic theory. Lucas summarized his critique:

"Given that the structure of an econometric model consists of optimal decision rules of economic agents, and that optimal decision rules vary systematically with changes in the structure of series relevant to the decision maker, it follows that any change in policy will systematically alter the structure of econometric models."

The Lucas critique is, in essence, a negative result. It tells economists, primarily, how not to do economic analysis. The Lucas critique suggests that if we want to predict the effect of a policy experiment, we should model the "deep parameters" (relating to preferences, technology, and resource constraints) that are assumed to govern individual behavior: so-called "microfoundations." If these models can

account for observed empirical regularities, we can then predict what individuals will do, taking into account the change in policy, and then aggregate the individual decisions to calculate the macroeconomic effects of the policy change.

Shortly after the publication of Lucas's article, Kydland and Prescott published the article "Rules rather than Discretion: The Inconsistency of Optimal Plans", where they not only described general structures where short-term benefits are negated in the future through changes in expectations, but also how time consistency might overcome such instances. That article and subsequent research led to a positive research program for how to do dynamic, quantitative economics.

The Lucas critique was an important methodological innovation. It does not invalidate that fiscal policy may be countercyclical, which some associate with John Maynard Keynes.”

Here follows an excerpt from the Wikipedia article on “reduced form.”

“In statistics, and particularly in econometrics, the *reduced form* of a system of equations is the result of solving the system for the endogenous variables. This gives the latter as functions of the exogenous variables, if any. In econometrics, the equations of a *structural form* model are estimated in their theoretically given form, while an alternative approach to estimation is to first solve the theoretical equations for the endogenous variables to obtain reduced form equations, and then to estimate the reduced form equations.

Let Y be the vector of the variables to be explained (endogenous variables) by a statistical model and X be the vector of explanatory (exogeneous) variables. In addition, let ε be a vector of error terms. Then the general expression of a structural form is $f(Y, X, \varepsilon) = 0$, where f is a function, possibly from vectors to vectors in the case of a multiple-equation model. The reduced form of this model is given by $Y = g(X, \varepsilon)$, with g a function.”

The structural form of a model is the form that arises naturally, describing the interrelationships among the variables in a system of equations. A linear structural model has the general form

$$By_t + \Gamma x_t = u_t$$

where B and Γ are $(n \times n)$ and $(n \times m)$ matrices of parameters (coefficients), u_t is an $(n \times 1)$ vector of model error terms, x_t is an $(m \times 1)$ vector of all of the predetermined (exogenous) variables, and y_t is an $(n \times 1)$ vector of all of the endogenous variables. The statement that x_t are predetermined is taken to mean that the x_t are uncorrelated with the model error terms, $E(x_t u_t') = \mathbf{0}$.

Assuming that B is invertible (positive definite), we may write this model in the form

$$y_t = -B^{-1}\Gamma x_t + B^{-1} u_t = \Pi' x_t + v_t$$

where $\Pi' = -B^{-1}\Gamma$ and $v_t = B^{-1} u_t$.

This second form of the model, which expresses the endogenous variates solely as a function of the exogenous variates, is called the reduced form of the model.

The primary advantage of the structural form is that it easier to understand the relationships among the model variables. A disadvantage of the structural form of the model is that (if there is

more than one endogenous variable) ordinary-least-squares estimates of the model parameters are biased.

The standard form of a VAR mode of order p , is

$$\Phi(B)(\mathbf{Z}_t - \boldsymbol{\mu}) = \mathbf{a}_t$$

where $\Phi(B) = I - \Phi_1 B - \Phi_2 B^2 - \dots - \Phi_p B^p$, Φ_i is a $(k \times k)$ parameter matrix, and \mathbf{a}_t is a white-noise sequence with mean $\mathbf{0}$ and covariance matrix Σ . This model may be written as

$$(\mathbf{Z}_t - \boldsymbol{\mu}) = \sum_{j=1}^p \Phi_j (\mathbf{Z}_{t-j} - \boldsymbol{\mu}) + \mathbf{a}_t$$

Since the \mathbf{a}_t are uncorrelated with the \mathbf{Z}_{t-j} , this model representation is in reduced form. A VAR is put into structural form as follows. (See p. 520 of Box-Jenkins book for details.)

Since the matrix $\Sigma = E[\mathbf{a}_t \mathbf{a}_t']$ is (assumed to be) positive definite there exists a lower-triangular matrix $\Phi_0^\#$ with ones on the diagonal such that $\Phi_0^\# \Sigma \Phi_0^\# = \Sigma^\#$ is a diagonal matrix with positive diagonal elements. Premultiplying the above equation by $\Phi_0^\#$ produces

$$\Phi_0^\# (\mathbf{Z}_t - \boldsymbol{\mu}) = \sum_{j=1}^p \Phi_j^\# (\mathbf{Z}_{t-j} - \boldsymbol{\mu}) + \mathbf{b}_t$$

where $\Phi_j^\# = \Phi_0^\# \Phi_j$ and $\mathbf{b}_t = \Phi_0^\# \mathbf{a}_t$ with $cov[\mathbf{b}_t] = \Sigma^\#$.

In many applications, the predetermined variables have substantive meaning, such as attributes of a physical entity. In the preceding, the predetermined variables were values of the observed variate at times prior to time t . In some applications, a structural variable may not have physical meaning, but may

simply be a geometrical construct, such as a trend observed in a time series. The term “structural model” may refer to models that include variables of either kind.

The books *Forecasting, structural time series models and the Kalman filter* by Andrew C. Harvey and *Time Series Analysis by State Space Methods* 2nd ed. By J. Durbin and S. J. Koopman discuss structural models in detail. ARIMA models remove levels and trends; the structural variables in those models have substantive meaning. As an example, the “airline ticket sales” example may be formulated as a model involving the single observed variable, or it may be represented as a structural model with an explicit trend and seasonal component. Both representations lead to accurate forecasts. In some applications, it is desired to estimate features of a time series, such as a trend or seasonal component. In those situations, the structural approach is preferred.

Econometric Models

Application of statistical methodology in the field of economics is referred to as econometrics. The field of econometrics applies the same models discussed here. In general, however, a statistical model in an economics application would involve greater consideration of structural models expressed in terms of economic variables.

An Internet search using terms such as “econometrics in R” will identify R resources relating to econometrics. Examples include *Econometrics in R* by Grant V. Farnsworth, October 26, 2008 , posted at <https://cran.r-project.org/doc/contrib/Farnsworth-EconometricsInR.pdf> ; and *Using R to Teach Econometrics* by Jeff Racine and Rob Hyndman, posted at <https://robjhyndman.com/papers/R.pdf> Here follows the abstract and initial part of the introduction from the latter article:

“R, an open-source programming environment for data analysis and graphics, has in only a decade grown to become a de-facto standard for statistical analysis against which many popular commercial programs may be measured. The use of R for the teaching of econometric methods is appealing. It provides cutting-edge statistical methods which are, by R's open-source nature, available immediately. The software is stable, available at no cost, and exists for a number of platforms, including various flavors of Unix and Linux, Windows(9x/NT/2000), and the MacOS. Manuals are also available for download at no cost, and there is extensive on-line information for the novice user. This review focuses on using R for teaching econometrics. Since R is an extremely powerful environment, this review should also be of interest to researchers.

R: an overview. There is a wide variety of programs for conducting statistical analysis. Many are powerful commercial products, such as Gauss, MATLAB, Minitab, SAS, Shazam, Stata, SPSS, S-PLUS, and TSP. Unfortunately, commercial software can be costly, sometimes prohibitively so, can involve relatively long development cycles, and rarely features experimental or cutting-edge statistical methods. Fortunately, there now exists an open source alternative to commercial statistical programs that is available for a variety of computing platforms including Windows, MacOS, MacOSX, FreeBSD, NetBSD, Linux, IRIX, Solaris, OSF/1, AIX and HP-UX. This software has been developed by a core team of leading statisticians and programmers. The software is called `R' and is available under the GNU Public License. To obtain the software, simply go to the site <http://www.r-project.org> and follow the download instructions.”

Microsimulation Models

From the Wikipedia article on microsimulation (or microanalytic simulation), posted at Internet website <https://en.wikipedia.org/wiki/Microsimulation> , is the following definition:

“The International Microsimulation Association,^[1] defines microsimulation as a modelling technique that operates at the level of individual units such as persons, households, vehicles or firms. Within the model each unit is represented by a record containing a unique identifier and a set of associated attributes – e.g. a list of persons with known age, sex, marital and employment status; or a list of vehicles with known origins, destinations and operational characteristics. A set of rules (transition probabilities) are then applied to these units leading to simulated changes in state and behaviour. These rules may be deterministic (probability = 1), such as changes in tax liability resulting from changes in tax regulations, or stochastic (probability ≤ 1), such as chance of dying, marrying, giving birth or moving within a given time period. In either case the result is an estimate of the outcomes of applying these rules, possibly over many time steps, including both total overall aggregate change and (importantly) the way this change is distributed in the population or location that is being modeled.”

In this article, we shall restrict attention to econometric microsimulation, which is concerned with the behavior or individuals or families over time.

In the microsimulation approach to forecasting, a large sample of households is taken from a household survey, and used as a basis for estimating quantities that are dependent on household

characteristics, such as welfare caseloads and budgets. A detailed description of the process is presented in *Policy Exploration through Microanalytic Simulation*, by Guy Orcutt, Steven Caldwell and Richard Wertheimer II (Urban Institute, 1976).

The Urban Institute currently maintains a microsimulation model called the Transfer Income Model, version 3, TRIM3.

An R package that performs microsimulation is MicSim, described in the article, “The MicSim Package of R: An Entry-Level Toolkit for Continuous-Time Microsimulation” by Sabine Zinn (*International Journal Of Microsimulation* (2014) 7(3) 3-32 International Microsimulation Association) posted at Internet website

https://www.microsimulation.org/IJM/V7_3/2_IJM_7_3_Zinn.pdf .

4. Deterministic Models (and Primarily Deterministic Models)

Computable general equilibrium models (CGE models)

The following description of computable general equilibrium models is quoted from the Wikipedia article on that topic:

“Computable general equilibrium (CGE) models are a class of economic models that use actual economic data to estimate how an economy might react to changes in policy, technology or other external factors. CGE models are also referred to as AGE (applied general equilibrium) models.

Overview

A CGE model consists of equations describing model variables and a database (usually very detailed) consistent

with these model equations. The equations tend to be neo-classical in spirit, often assuming cost-minimizing behaviour by producers, average-cost pricing, and household demands based on optimizing behaviour. However, most CGE models conform only loosely to the theoretical general equilibrium paradigm. For example, they may allow for:

- non-market clearing, especially for labour (unemployment) or for commodities (inventories)
- imperfect competition (e.g., monopoly pricing)
- demands not influenced by price (e.g., government demands)

A CGE model database consists of:

tables of transaction values, showing, for example, the value of coal used by the iron industry. Usually the database is presented as an input-output table or as a social accounting matrix (SAM). In either case, it covers the whole economy of a country (or even the whole world), and distinguishes a number of sectors, commodities, primary factors and perhaps types of household. Sectoral coverage ranges from relatively simple representations of capital, labor and intermediates to highly-detailed representations of specific sub-sectors (e.g., the electricity sector in GTAP-Power).

elasticities: dimensionless parameters that capture behavioural response. For example, export demand elasticities specify by how much export volumes might fall if export prices went up. Other elasticities may belong to the constant elasticity of substitution class. Amongst these are Armington elasticities, which show whether products of different countries are close substitutes, and elasticities measuring how easily inputs to production may be substituted for one another. Income elasticity of demand

shows how household demands respond to income changes.

CGE models are descended from the input-output models pioneered by Wassily Leontief, but assign a more important role to prices. Thus, where Leontief assumed that, say, a fixed amount of labour was required to produce a ton of iron, a CGE model would normally allow wage levels to (negatively) affect labour demands.

CGE models derive too from the models for planning the economies of poorer countries constructed (usually by a foreign expert) from 1960 onwards . Compared to the Leontief model, development planning models focused more on constraints or shortages—of skilled labour, capital, or foreign exchange.

CGE modelling of richer economies descends from Leif Johansen's 1960 MSG model of Norway, and the static model developed by the Cambridge Growth Project in the UK. Both models were pragmatic in flavour, and traced variables through time. The Australian MONASH model is a modern representative of this class. Perhaps the first CGE model similar to those of today was that of Taylor and Black (1974).

CGE models are useful whenever we wish to estimate the effect of changes in one part of the economy upon the rest. For example, a tax on flour might affect bread prices, the CPI, and hence perhaps wages and employment. They have been used widely to analyse trade policy. More recently, CGE has been a popular way to estimate the economic effects of measures to reduce greenhouse gas emissions.

CGE models always contain more variables than equations—so some variables must be set outside the model. These variables are termed exogenous; the remainder, determined by the model, are called endogenous. The choice of which variables are to be exogenous is called the model closure, and may give rise to controversy. For example, some modelers hold employment and the trade balance fixed; others allow these to vary. Variables defining technology, consumer tastes, and government instruments (such as tax rates) are usually exogenous.

Today there are many CGE models of different countries. One of the most well-known CGE models is global: the GTAP model of world trade.

CGE models are useful to model the economies of countries for which time series data are scarce or not relevant (perhaps because of disturbances such as regime changes). Here, strong, reasonable, assumptions embedded in the model must replace historical evidence. Thus developing economies are often analysed using CGE models, such as those based on the IFPRI template model.”

Here follows a query, “[R] Computable General Equilibrium (CGE) models in R,” posted on August 11, 2010, at <https://tolstoy.newcastle.edu.au/R/e11/help/10/08/4395.html> , asking whether an R package existed to implement computable general equilibrium models.

“Has someone implemented a Computable General Equilibrium (CGE) model in R? Usually such models are coded in GAMS (a general algebraic solver) or GEMPACK (a solver specifically for CGE models). As it is possible to implement CGE models in the econometric software EViews

(Essama-Nssah, B. (2004), Building and Running General Equilibrium Models in EViews, <http://ssrn.com/paper=636617>), I suspect it can also be done in R. I am aware of the BB package for solving a large system of nonlinear equations (<http://cran.r-project.org/web/packages/BB/>), but what I am looking for is examples of CGE models coded in R. -- Luc Hens”

From the lack of response to this query, it appears that the major software available for implementing CGEs consists of GAMS, GEMPACK and EViews.

A summary of the technical basis for CGE follows.

A general equilibrium model describes a stable state of an economy at a high level of aggregation. The model represents the flow of money between sectors of the economy. In the equilibrium state, it is assumed that individuals and firms allocate their monetary resources to maximize utility, that the prices of goods and services are the same for individuals and firms, and that the prices are determined so that all produced goods are consumed.

The basic entities of a general equilibrium model are individuals, firms, governments, resources, goods and services. The model variables describe these entities, such as amounts of a resource, prices, and tax rates. Model parameters specify characteristics of model entities and functional relationships between the model entities.

A computable general equilibrium model is one for which data are available to enable estimation of quantities of interest, such as the new allocation of resources if changes are made to model variables or parameters, such as changes in tax rates, constraints on imports, or transfer payments.

A major component of a CGE is a social accounting matrix (SAM), that specifies the transfers of money between model entities (individuals, firms, governments in an economy) in a particular year or years. We shall restrict attention to static CGEs, in which attention is restricted to a single “base” year, not to dynamic CGEs that include data for several years. Social accounting matrices are available for many countries for several base years.

A social accounting matrix is similar to an input-output table. The input-output table is limited to a description of the production and consumption in the producing sectors of an economy. A SAM includes other economic aspects, such as the government sector, taxes, savings and investment.

A CGE involves a large amount of data about the economy and about the response of individuals and firms to changes in variables such as prices and resource availability. The responses to changes are described by elasticities (the ratio of a percentage change in one variable to the percentage change in another variable). A major problem facing implementation of CGEs is the limited availability of data describing consumer preferences and elasticities. A standard approach to estimating the parameters of utility functions and production functions is to assume parametric functional forms for them and “calibrate” values for the parameters that are consistent with the SAM data.

A major decision about a CGE application is whether the equations defining the model are defined in terms of nonlinear equations involving levels of variables or linear equations involving percentages of variables. If defined in terms of levels and nonlinear equations, it is necessary to estimate the parameters of those equations from the base-year data (“calibration”).

Some applications of CGEs have involved the use of econometric models to estimate model parameters, but this activity has been limited.

Although CGEs are used to make forecasts about the effects of changes in policy variables, the justification for this is somewhat tenuous. CGE models involve many parameters, and there is often little justification for the assumed values of the parameters. A legitimate use of CGEs is to conduct sensitivity analysis to estimate the sensitivity of output variables of interest to changes of input variables of interest, conditional on the model specification. This is similar to making deterministic population projections based on assumptions about birth, death and migration rates, without making probability statements about the projections.

Detailed descriptions of CGE are presented in the following references:

Burfisher, Mary E., *Introduction to Computable General Equilibrium Models*, 2nd ed., Cambridge University Press, 2016

Hosoe, Nobuhiro, Kenji Gasawa and Hideo Hashimoto, *Textbook of Computable General Equilibrium Modeling, Programming and Simulations*, Palgrave Macmillan / St. Martin's Press, 2010

The first reference focuses on the use of linear models (such as GEMPACK and GTAP), and the second on the use of nonlinear models (solvable with GAMS).

There is much reference material about CGEs on the Internet, including introductory material and detailed examples. The key data requirements for a CGE are a social accounting matrix and

elasticities (which describe the behavioral response to changes in prices or resource amounts). Constructing estimates of these quantities involves a substantial amount of effort. An Internet search reveals a number of articles describing procedures for constructing a SAM.

The decision to use a CGE as the basis for estimation of the impact of a policy change would not rest on a single application, but on a decision to establish an institutional capacity to perform CGE analyses for many applications in the future. Whereas the time-series approach to forecasting can be implemented quickly, that is not true of CGE estimation.

While the time-series models discussed earlier could be described as a “statistical” approach to forecasting, and the CGE approach described as an “optimization” approach, that characterization is somewhat misleading, since estimation methods based on time-series estimation approaches usually involve optimization (e.g., minimization of an error sum of squares, or maximization of a likelihood function). On the other hand, referring to a CGE as a “deterministic” approach is also a little misleading, since estimation of the model parameters (SAM entries, elasticities, utility and production function parameters) could involve statistical analysis.

Population-based (demographic, regional, small-area-estimation) models

A powerful method for forecasting phenomena that are closely related to population is to use a cohort-component population projection model to project population into the future, and then use a model-based-estimation approach (such as synthetic estimation or small-area estimation) to construct forecasts.

This topic could have been included in the “Stochastic Models” section, since it involves statistical techniques and estimates. Since the errors in judgment about the population projections are often substantially greater than errors associated with estimation conditional on the population projections, it is categorized under deterministic rather than stochastic models.

An Internet search of terms such as “R for demography,” “R for small-area estimation,” “R for model-based estimation,” “R for synthetic estimation,” “R for population projections,” “R for population forecasting,” “R for demographic forecasting” and the like will show that R packages are available to perform model-based estimation. Examples of packages are “sae: An R Package for Small Area Estimation” by Isabel Molina and Yolanda Marhuenda posted at <https://journal.r-project.org/archive/2015/RJ-2015-007/RJ-2015-007.pdf> and “The R Package emdi for Estimating and Mapping Regionally Disaggregated Indicators” by Ann-Kristin Kreutzmann, Sören Pannier, Natalia Rojas-Perilla, Timo Schmid, Matthias Templ and Nikos Tzavidis posted at https://cran.r-project.org/web/packages/emdi/vignettes/vignette_emdi.pdf.

The two principal sources of R programs for demographic forecasting are Prof. Rob Hyndman’s R package, “Package ‘demography’,” April 22, 2019, posted at <https://cran.r-project.org/web/packages/demography/demography.pdf> and the software package, “YourCast: Time Series Cross-Sectional Forecasting with Your Assumptions,” by Profs. Federico Girosi and Gary King, posted at <https://gking.harvard.edu/yourcast> (formerly an R package). *YourCast* implements the methods for demographic forecasting discussed in *Demographic Forecasting* by Federico Girosi and Gary King, Princeton University Press, 2008. It is posted at <http://gking.harvard.edu/files/abs/smooth-abs.shtml>. The author notes, “Please read at least Chapter 1 of the book before attempting to use *YourCast*.”

Demographic constructs such as mortality tables involve many correlated variables. The standard way of forecasting them is to apply the method of principal components (i.e., form a singular value decomposition of a matrix, forecast the eigenvalues (as a multivariate vector) using an ARIMA-type model (e.g., a VAR), and construct a forecasted matrix from the forecasted eigenvector. The method forecasting mortality using principal components is called the Lee-Carter model, after the men who popularized the use of the method in demographic forecasting.

A program for making synthetic-estimate forecasts based on population projections is the DESTINY software program package, developed by the author (see citation in References for access to source code).

There are two approaches to estimation of populations disaggregated by race or region. One is to construct demographic tables by those variables, and the other is to use the methodology of small-area estimation, such as synthetic estimation or other indirect-estimation technique. See the book, *Small Area Estimation* by J. N. K. Rao for detailed information on small-area estimation.

Cost-Benefit Models

An Internet search of “R for cost-benefit analysis” or “R for cost-benefit models” finds R software for cost-effectiveness analysis, but not for cost-benefit analysis. While this methodology is useful for predicting the effects of program and policy changes, it is evidently not sufficiently well-structured to warrant the development of special-purpose software to assist the analysis.

Anticipation Surveys

No R software was identified for specifically assisting the conduct of anticipation surveys. Considerable R software is available for analysis of sample survey data. A source that identifies some resources in this area is “Survey analysis in R” posted at <http://r-survey.r-forge.r-project.org/survey/> .

This topic could have been included in the “Stochastic Models” section, but for the fact that anticipation surveys are often ad-hoc, based on “judgment samples,” rather than designed sample surveys.

Content Analysis

Some R software is available for assisting content analysis. Discussion of this topic is presented in the article “Text Analysis in R” by Kasper Welbers, Wouter van Atteveldt and Kenneth Benoit (*Communication Methods And Measures* 2017, Vol. 11, No. 4, 245–265 <https://doi.org/10.1080/19312458.2017.1387238>) posted at https://kenbenoit.net/pdfs/text_analysis_in_R.pdf .

See also *ReadMe: Software for Automated Content Analysis* on Gary King’s website, <https://gking.harvard.edu/readme> .

Appendix A. Additional Discussion of the Univariate Univariable ARIMA Model

Additional discussion

For an ARIMA model, the forecasts may be computed directly from the model equation.

Let us denote the general autoregressive polynomial

$$\Phi(B)(1-B)^d$$

as

$$\Phi(B)(1-B)^d = \phi(B) = (1 - \phi_1 B - \dots - \phi_{p+d} B^{p+d}).$$

In what follows, for simplicity, we shall drop the mean term ϕ_0 .

Then the model may be written as

$$\phi(B)z_t = \Theta(B)a_t$$

or

$$z_t = \phi_1 z_{t-1} + \dots + \phi_{p+d} B^{p+d} z_{t-p-d} + a_t - \theta_1 a_{t-1} - \dots - \theta_q a_{t-q}$$

The forecast is calculated directly from this formula, substituting known or forecasted values for the z 's, and estimated values or zeros for the a 's. The value zero is used for future a 's. Past a 's are estimated from the equation

$$a_t = z_t - z_t^f,$$

where z_t^f is the forecasted value of z_t from the time $t-1$.

The a_t 's are the one-step-ahead forecast errors. For models involving moving-average parameters (θ s), in order to estimate the forecast from time t using the model formula, it is necessary to begin forecasting a few steps before the end of the series, so that estimates of the a_t 's will be available for the forecast from time t .

For example, for the model constructed above,

$$z_t = z_{t-1} + a_t - .05336 a_{t-1},$$

the forecast of the last observation in the time series, from the next-to-last time $t = 239$ is

$$Z_{240} = Z_{239} - .05336 a_{239},$$

where $z_{240} = 1.12388$, $z_{239} = 1.12386$ and a_{239} is estimated as its expected value, zero (since we have not estimated it from earlier forecasts).

This yields the forecast

$$Z_{240\text{estfrom239}} = 1.12386 - .05336 (0) = 1.12386.$$

and the estimated value of a_{240} as

$$a_{240\text{est}} = Z_{240\text{true}} - Z_{240\text{estfrom239}} = 1.12388 - 1.12386 = .00002.$$

Using this estimated value of a_{240} , we calculate the one-step-ahead forecast from time $t = 240$ as

$$Z_{241\text{estfrom240}} = Z_{240} - .05334 a_{240\text{est}} = 1.12388 - .05334(.00002) = 1.123879.$$

This value is in agreement with the program output.

Formulas for the Forecast Function and Forecast Error Variance

The following results show why the forecast may be calculated from the model equation, and derives a formula for the error variance of the forecasts. These results are from pp. 126-129 of the first edition of the Box-Jenkins book.

The standard (canonical) form of an ARIMA model is

$$\phi(B)z_t = \Theta(B)a_t$$

where

$$\phi(B) = \Theta(B)(1 - B)^d,$$

where $\Phi(B)$ and $\Theta(B)$ are polynomial of finite degrees p and q , respectively.

If the roots of the Φ polynomial lie outside the unit circle, and if $d=0$, then the process is stationary, and may be written in the form

$$z_t = \phi^{-1}(B)\Theta(B) a_t = \Psi(B) a_t$$

where

$$\Psi(B) = (1 - \psi_1 B - \psi_2 B^2 - \dots).$$

The function is called the transfer function or impulse response function of the model, and the coefficients ψ_1, ψ_2 , are called impulse responses.

The polynomial $\phi^{-1}(B)$ is the operator inverse of the polynomial $\phi(B)$. For example, if

$$\phi(B) = 1 - \phi_1 B$$

where ϕ_1 is less than one in magnitude, then

$$\phi^{-1}(B) = 1 + \phi_1 B + \phi_1^2 B^2 + \dots$$

If the roots of the Θ polynomial lie outside the unit circle, then the process is invertible, and may be written in the form

$$a_t = \Theta^{-1}(B)\phi(B) z_t = \Pi(B) z_t$$

where

$$\Pi(B) = 1 - \pi_1 B - \pi_2 B^2 - \dots$$

The polynomials Ψ and Π satisfy

$$\Psi(B) = \Pi^{-1}(B).$$

Since

$$\Psi(B) = \phi^{-1}(B)\Theta(B),$$

It follows that

$$\phi(B)\Psi(B) = \Theta(B),$$

and the values of the ψ s may be determined by equating the coefficients on powers of B on opposite sides of the preceding equation.

Since

$$\Pi(B) = \Theta^{-1}(B)\phi(B),$$

It follows that

$$\Theta(B)\Pi(B) = \phi(B),$$

and the values of the π s may be determined by equating the coefficients on powers of B on opposite sides of the preceding equation.

Hence we see that, if a model is stationary and invertible, then it may be represented in three alternative forms. The standard (canonical) form is the most efficient representation, since it is specified by a small number of parameters ($p + q$ of them). It is a useful form for determining the forecast function. The transfer-function form of the model,

$$z_t = \phi^{-1}(B)\Theta(B) a_t = \Psi(B) a_t$$

is useful in determining a formula for the variance of the forecast errors.

If a model has roots on the unit circle, then the operator $\Psi(B)$ cannot be inverted, and the alternative model representation $z_t = \Psi(B) a_t$ does not exist. Nevertheless, the formulas for the ψ s obtained by setting coefficients equal may still be used to obtain a formula for the variance of the forecast errors.

With these preliminary results, we shall now proceed to derive formulas for the forecast function and the variance of the forecast error variance. We shall present results for the case in which $d = 0$ and the roots of the ϕ and Θ polynomials lie outside the unit circle. In this case, the process is invertible and stationary. (The formula for the forecast error variance still holds for the case $d = 0$, but they are not proved here.)

Since the process is stationary, the $\phi(B)$ operator is invertible, and the process may be represented as

$$z_t = \phi^{-1}(B) a_t.$$

We shall write this representation as

$$z_t = \Psi(B) a_t.$$

Suppose that we are interested in forecasting the value of z_t r periods ahead of current time t (observations are available for z_t, z_{t-1}, \dots). This value is

$$z_{t+r} = \Psi(B)a_{t+r} = \sum_{j=0}^{\infty} \psi_j a_{t+r-j}$$

where the value of ψ_1 is 1. (The function $\Psi(B)$ is called the transfer function, and the values ψ_1, ψ_2, \dots are called impulse responses.) The values of the ψ 's may be obtained by equating coefficients in the equation

$$\varphi(B)(1 + \psi_1 + \dots) = \theta(B).$$

Suppose that it is desired to estimate the value of z_{t+r} from current time t , as a linear function of the available observations, z_t, z_{t-1}, \dots . Denote this estimate, or forecast, as $\hat{z}_t(r)$. Since the z 's are linear functions of the a 's, the forecast may be written as a linear function of the a 's.

Write this forecast as

$$\hat{z}_t(r) = \psi_1^* a_1 + \psi_2^* a_2 + \dots,$$

where the coefficients ψ_i^* are to be determined to minimize the mean squared error of prediction,

$$\begin{aligned} E[z_{t+r} - \hat{z}_t(r)]^2 \\ = (1 + \psi_1^2 + \dots + \psi_{r-1}^2) \sigma_a^2 + \sum_{j=0}^{\infty} \{\psi_{r+j} - \psi_{r+j}^*\}^2 \sigma_a^2. \end{aligned}$$

The minimizing values are $\psi_{r+j}^* = \psi_{r+j}$. It follows that

$$z_{t+r} = (a_{t+r} + \psi_1 a_{t+r-1} + \dots + \psi_{r-1} a_{t+1}) + (\psi_r a_t + \psi_{r+1} a_{t-1} + \dots) = e_t(r) + \hat{z}_t(r)$$

where $e_t(r)$ denotes the error of the forecast $\hat{z}_t(r)$ at lead time r .

The conditional expectation of z_{t+r} given the z 's up to time t is

$$E(z_{t+r} | z_t, z_{t-1}, \dots) = E(z_{t+r} | a_t, a_{t-1}, \dots) = \psi_r a_t + \psi_{r+1} a_{t-1} + \dots = \hat{z}_t(r).$$

Considered as a function of r for fixed t , the function $\hat{z}_t(r)$ is called the *forecast function* for origin t .

The mean and variance of the forecast error for lead time r are

$$E[e_t(r)] = 0$$

And

$$\text{var}[e_t(r)] = (1 + \psi_1^2 + \psi_2^2 + \dots + \psi_{r-1}^2) \sigma_a^2.$$

The one-step-ahead forecast errors are

$$e_t(1) = z_{t+1} - \hat{z}_t(1) = a_{t+1}.$$

As discussed earlier, to determine the variance for the particular model

$$\phi(B) z_t = \theta(B) a_t$$

we must determine the values of the ψ 's by equating coefficients on both sides of the equation

$$\varphi(B)(1 + \psi_1 + \dots) = \theta(B).$$

For the model

$$(1 - \phi B) z_t = (1 - \theta B) a_t$$

we have

$$(1 - \phi B)(1 + \psi_1 B + \dots) = (1 - \theta B).$$

Setting the coefficients of the powers of B on the two sides of the equation equal, we obtain

$$-\phi + \psi_1 = -\theta$$

$$\phi \psi_1 - \psi_2 = 0$$

$$\phi \psi_2 - \psi_3 = 0$$

...

or

$$\psi_1 = \phi - \theta$$

$$\psi_2 = \phi(\phi - \theta)$$

$$\psi_3 = \phi^2(\phi - \theta)$$

....

For the model estimated for the EUR/USD one-minute data, $\phi = 1$ and $\theta = .0534$. Hence

$$\psi_1 = \phi - \theta = 1 - .0534 = .9466$$

$$\psi_2 = \phi(\phi - \theta) = .9466$$

$$\psi_3 = \phi^2(\phi - \theta) = .9466$$

....

Hence the variance of the error in the r-ahead forecast is

$$V(r) = (1 + (r-1).9466^2)\sigma_a^2.$$

For values of r from 1 to 4, the values of the coefficient of σ_a^2 are 1, 1.8961, 2.794, and 3.690.

Since σ_a^2 is estimated as 8.439229e-09, σ_a is estimated as the square root .00009186528. The square roots of the variance coefficients are 1, 1.377, 1.671, and 1.920. Multiplying these factors by .000091865 yields the following standard errors for the forecasts from lead times 1 through 4: .000091865, .0001265, .0001535, and .0001764.

This matches the program output.

References

ARIMA-type Statistical Time Series Analysis

Box, G. E. P., and Gwilym Jenkins, *Time Series Analysis, Forecasting Control*, first edition Holden-Day, 1970, latest edition is 5th edition by George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel and Greta M. Ljung, Wiley, 2016.

Tsay, Ruey S., *Multivariate Time Series Analysis with R and Financial Applications*, Wiley, 2014.

Lütkepohl, Helmut, *New Introduction to Multiple Time Series Analysis*, Springer, 2006

Hamilton, James D., *Time Series Analysis*, Princeton University Press, 1994

Cryer, Jonathan D. and Kung-Sik Chan, *Time Series Analysis with Applications in R*, Springer, 2008 (univariate)

Structural-Type Time Series Analysis, Emphasizing Kalman Filtering

Harvey, Andrew C., *Forecasting, structural time series models and the Kalman filter*, Cambridge University Press, 1989

Durbin, J. and S. J. Koopman, *Time Series Analysis by State Space Methods*, 2nd edition, Oxford University Press, 2012

Causal Inference

Pearl, Judea, *Causality: Models, Reasoning, and Inference*, 2nd Edition, Cambridge University Press, 2009 (1st Ed. 2000)

Morgan, Stephen L. and Christopher Winship, *Counterfactuals and Causal Inference: Methods and Principles for Social Research*, Cambridge University Press, 2007, 2nd ed. 2014

Econometrics

Wooldridge, Jeffrey M., *Econometric Analysis of Cross Section and Panel Data*, 2nd Ed., The MIT Press, 2010 (1st Ed. 2002).

Greene, William H., *Econometric Analysis*, 7th Edition, Prentice Hall, 2012

Computable General Equilibrium Models

Burfisher, Mary E., *Introduction to Computable General Equilibrium Models*, 2nd ed., Cambridge University Press, 2016

Hosoe, Nobuhiro, Kenji Gasawa and Hideo Hashimoto, *Textbook of Computable General Equilibrium Modeling, Programming and Simulations*, Palgrave Macmillan / St. Martin's Press, 2010

Thomas W. Hertel, ed., *Global Trade Analysis, Modeling and Applications*, Cambridge University Press, 1997

ten Raa, Thijs, *Input-Output Analysis*, Cambridge University Press, 2005

Demography

Girosi, Federico and Gary King, *Demographic Forecasting*, Princeton University Press, 2008

Caldwell, Joseph George, *DESTINY 2005 Demographic Estimation, Forecasting and Analysis System, Description of Capabilities International Version 3.0.01*, December 26, 2005 posted at <http://foundationwebsite.org/DestCapINTL.htm> and *DESTINY 2005 Demographic Estimation, Forecasting and Analysis System, User's Manual International Version 3.0.01*, December 26, 2005 posted at <http://foundationwebsite.org/DestUserManINTL.htm> .

Small-Area Estimation

Rao, J. N. K., *Small Area Estimation*, Wiley, 2003

R

Adler, Joseph, *R in a Nutshell* 2nd ed., O'Reilly, 2012

Crawley, Michael J., *The R Book*, Wiley, 2007

Sawitzki, Günther, *Computational Statistics, An Introduction to R*, CRC Press, 2009

Cornillon, Pierre-Andre, Arnaud Guyader, François Hussen, Nicolas Jégou, Julie Josse, Maela Kloareg, Eric Matzner-Løber and Laurent Rouvière, *R for Statistics*, CRC Press, 2012

FndID(19)

FndTitle(A Survey of Methods for Forecasting, Policy Analysis and Planning, with Examples in R)

FndDescription(This article presents examples of a variety of methods used for forecasting and policy analysis. The primary purpose is to show that the methods can be implemented using data and software that are freely available on the Internet.)

FndKeywords(R; forecasting; policy analysis; planning; causal inference; stochastic models, time series analysis; Box-Jenkins; ARIMA; VAR; computable general equilibrium; CGE, demographic projections; population projections; small-area statistics; microsimulation; Kalman filter; Bayesian statistics)