Journal of Statistical Software

May 2011, Volume 41, Issue 7. *http://www.jstatsoft.org/*

REGCMPNT – A Fortran Program for Regression Models with ARIMA Component Errors

William R. Bell
U.S. Census Bureau

Abstract

RegComponent models are time series models with linear regression mean functions and error terms that follow ARIMA (autoregressive-integrated-moving average) component time series models. Bell (2004) discusses these models and gives some underlying theoretical and computational results. The **REGCMPNT** program is a Fortran program for performing Gaussian maximum likelihood estimation, signal extraction, and forecasting with RegComponent models. In this paper we briefly examine the nature of RegComponent models, provide an overview of the **REGCMPNT** program, and then use three examples to show some important features of the program and to illustrate its application to various different RegComponent models.

Keywords: RegComponent model, time series, unobserved components, time series software.

1. Introduction

REGCMPNT is a Fortran program for Gaussian maximum likelihood (ML) estimation, signal extraction, and forecasting for univariate RegComponent models, which are time series models with linear regression mean functions and error terms following ARIMA (autoregressive-integrated-moving average) component time series models. Bell (2004) gives a general discussion of RegComponent models, presenting three examples, as well as discussing underlying theoretical and computational results for Gaussian ML estimation, forecasting, and signal extraction. The **REGCMPNT** program itself, along with example input and output files, is available along with this manuscript. A Windows interface is also under development and is expected to be available shortly from the author.

This paper illustrates the capabilities of the **REGCMPNT** program by showing in detail its use in several examples (Sections 4–6). Prior to this, Section 2 gives a brief overview of RegComponent models, and Section 3 discusses how to run the **REGCMPNT** program.

Section 4 then shows how to use **REGCMPNT** to fit the local level model (Commandeur, Koopman, and Ooms 2011, Equation 3) to the Nile riverflow data modeled in Durbin and Koopman (2001, Chapter 2). Section 5 shows how **REGCMPNT** can handle a seasonal structural model of Harvey (1989) that also includes regression terms for trading-day and Easter holiday effects. Section 6 shows how **REGCMPNT** can handle a model for a time series of repeated survey estimates whose sampling variances change over time. Finally, Section 7 offers some concluding remarks.

2. A brief overview of RegComponent models

The general form of a RegComponent model is

$$y_t = x_t^\top \beta + \sum_{j=1}^{m} h_{jt}\mu_t^{(j)}, \tag{1}$$

where

y_t is the observed time series with observations at time points $t = 1, \ldots, n$. Note that y_t may be a transformation (e.g., logarithms) of an original time series.

x_t is an $r \times 1$ vector of known regression variables and β is the corresponding vector of (fixed) regression parameters.

h_{jt} for $j = 1, \ldots, m$ are series of known constants that we call *scale factors*. Often $h_{jt} = 1$ for all j and t.

$\mu_t^{(j)}$ for $j = 1, \ldots, m$ are independent unobserved component series following ARIMA models.

A general notation for the ARIMA models for the $\mu_t^{(j)}$ in (1) is

$$\phi_j(B)\Delta_j(B)\mu_t^{(j)} = \theta_j(B)\zeta_{jt} \tag{2}$$

where $\phi_j(B)$, $\Delta_j(B)$, and $\theta_j(B)$ are the autoregressive (AR), "differencing," and moving-average (MA) operators, which are polynomials in the backshift operator B ($B\mu_t^{(j)} = \mu_{t-1}^{(j)}$). These polynomials can be multiplicative, as in seasonal ARIMA models. We require the $\phi_j(B)$ to have all their zeros outside the unit circle, and the $\theta_j(B)$ to have all their zeros on or outside the unit circle. Common versions of the $\Delta_j(B)$ would be (i) the identity operator ($\Delta_j(B) = 1$), corresponding to stationary components (such as the observation disturbance ϵ_t in Equation 1 of Commandeur *et al.* 2011); (ii) a nonseasonal $(1 - B)$ or seasonal $(1 - B^s)$ difference, or a product of these; or (iii) a seasonal summation operator, $1 + B + \cdots + B^{s-1}$ (see Equation 5 in Commandeur *et al.* 2011 or equation (7) in the model of Section 5 below). The $\Delta_j(B)$ typically have all their zeros on the unit circle, and usually must have no common zeros, as common zeros can create problems for signal extraction results (Bell 1984, 1991; Kohn and Ansley 1987). Exceptions to this rule occur for components $h_{jt}\mu_t^{(j)}$ whose h_{jt} are not all equal over t (as occurs for models with time-varying regression parameters.) The ζ_{jt} are *i.i.d.* $N(0, \sigma_j^2)$ (white noise) innovations, independent of one another (which implies $\mathrm{cov}(\zeta_{it}, \zeta_{jt'}) = 0$ unless $i = j$ and $t = t'$.)

Effect type	Comments
Constant term	Allows for nonzero mean levels in models with no differencing, and for trend constants in models with differencing.
Fixed seasonal	Modeled with either monthly (or quarterly) contrast variables or with trigonometric terms.
Trading-day	Variables for modeling trading-day effects in flow or stock series, as well as for modeling length-of-month (or quarter) effects or leap-year effects.
Holiday	Variables for modeling Easter, Labor Day, or Thanksgiving effects.
Outliers and interventions	Variables for modeling additive outliers, level shifts, and ramp effects.
User defined	May read in data for regression variables to model other effects.

Table 1: Regression effects in **REGCMPNT**.

If $m = 1$ and $h_{1t} = 1$ for all t, then model (1) reduces to the general RegARIMA model as a special case. RegARIMA stands for a regression model with error terms that follow an ARIMA model. See Bell and Hillmer (1983) and Findley, Monsell, Bell, Otto, and Chen (1998) for discussion of RegARIMA modeling. The **X-12-ARIMA** seasonal adjustment program (Findley *et al.* 1998; U.S. Census Bureau 2009) provides RegARIMA modeling capabilities that have much in common with the capabilities of the **REGCMPNT** program, and in fact the two programs share a lot of Fortran code.

Model (1) extends the pure ARIMA components model given as Equation 18 of Commandeur *et al.* (2011) in two ways. The first extension involves the regression mean function $x_t^\top \beta$ (also mentioned in Section 2.2 of Commandeur *et al.* 2011). **REGCMPNT** allows models to include regression variables for several types of regression effects commonly used in modeling seasonal economic time series. These are summarized in Table 1. They are substantially the same variables that are available in the **X-12-ARIMA** program (U.S. Census Bureau 2009), though **X-12-ARIMA** has a few extensions and modifications to the variables that are not currently included in **REGCMPNT**.

The second extension involves the scale factors h_{jt}. These enter the state space representation of the model (Equation 1 in Commandeur *et al.* 2011) through the matrix Z_t, since the first element of the state space representation of each ARIMA component $\mu_t^{(j)}$ can be taken to be $\mu_t^{(j)}$ itself. (Note discussion in Section 4 of Commandeur *et al.* 2011.) This is analogous to how regression effects (with constant or time-varying coefficients) enter the state space representation in Section 2.2 of Commandeur *et al.* (2011). Thus, as mentioned above and as discussed at the end of Section 5, one application of the scale factors h_{jt} in model (1) is to accommodate time-varying regression coefficients that follow ARIMA models (with the corresponding regression variables given by the associated h_{jt}'s).

Another important application of model (1) is to time series y_t obtained as estimates from a repeated sample survey. In this case we write

$$y_t = Y_t + e_t \tag{3}$$

where Y_t is the time series of true population characteristics being estimated by y_t, and $e_t = y_t - Y_t$ is the sampling error in y_t as an estimate of Y_t. In (3) the true series (or *signal component*), Y_t, includes any regression terms $x_t^\top \beta$, and so follows a RegComponent model, which could possibly be the special case of a RegARIMA model. The sampling error, e_t, is generally assumed to have mean zero (i.e., the y_t are assumed to be unbiased estimates of the Y_t), and we can assign e_t to the last component in (1), i.e., $e_t = h_{mt}\mu_t^{(m)}$, with $\mu_t^{(m)}$ generally assumed to follow a stationary ARMA model (no differencing). The h_{mt} then allow for the variance of e_t to vary over time (something fairly common in repeated surveys) by defining $h_{mt} = \sqrt{\mathrm{Var}(e_t)}$ and setting the innovation variance of $\mu_t^{(m)}$ so that $\mathrm{Var}(\mu_t^{(m)}) = 1$ for all t. An important point about application of RegComponent models to time series from repeated surveys is that the parameters of the model for e_t should be estimated using estimates of variances and autocovariances of e_t obtained from survey microdata. (See Wolter 1985 for discussion of survey variance estimation.) The parameters of the model for e_t are then held fixed when model (1) is estimated. The option to fix parameters of the ARIMA component models in (1) is a key feature of **REGCMPNT**.

Scott and Smith (1974) and Scott, Smith, and Jones (1977) first suggested use of time series modeling and signal extraction to improve estimates from repeated surveys. Further discussion covering the use of RegComponent models in this context, including examples analyzed with the **REGCMPNT** program, is given in Bell and Hillmer (1990) and Bell (2004).

Bell (2004) discusses ML estimation of RegComponent models, giving details for the case where all the scale factors are 1 ($h_{jt} \equiv 1$). To summarize, the **REGCMPNT** program maximizes the likelihood of a RegComponent model (1) via an iterative generalized least squares (IGLS) algorithm that alternates between (*i*) maximizing the log-likelihood over the regression parameters β for given values of the ARMA parameters and variances of the component models (2), and (*ii*) maximizing the log-likelihood over the unknown ARMA parameters and variances for a given value of β. The "unknown" ARMA parameters and variances are those not specified as fixed at particular values in the program's input file. Step (*i*) is achieved by generalized least squares regression of the differenced data ($\Delta(B)y_t$) on the differenced regression variables ($\Delta(B)x_{jt}$), where $\Delta(B) = \prod_{j=1}^{m} \Delta_j(B)$ is the overall differencing operator for the model. Step (*ii*) is achieved by computing regression residuals, $z_t = y_t - x_t^\top \beta$, and maximizing the log-likelihood for the unknown ARMA parameters and variances, where this is the log of the joint density of $\Delta(B)z_t$ for $t = d+1, \ldots, n$, where d is the order of $\Delta(B)$. For this step, the ARIMA component model for $z_t = \sum_{j=1}^{m} h_{jt}\mu_t^{(j)}$ is put in state space form and the Kalman filter (with a suitable initialization) is used to evaluate the log-likelihood. (This approach works generally, not just in the case where $h_{jt} \equiv 1$.) The maximization for step (*ii*) is carried out by the **MINPACK** Fortran routines (More, Garbow, and Hillstrom 1980). Commandeur *et al.* (2011) discuss the general use of the state space form and Kalman filter for likelihood evaluation. While their approach of putting the regression parameters in the state vector (Commandeur *et al.* 2011, Section 2.2) differs from the IGLS approach, both approaches would lead to the same ML estimates of the model parameters.

The "suitable initialization" of the Kalman filter referred to above is needed to deal with the nonstationarity resulting from the differencing in the ARIMA component models (2).

REGCMPNT uses the initialization of Bell and Hillmer (1991), which yields the "transformation approach" results of Ansley and Kohn (1985). Other approaches to these computations are possible (e.g., Koopman 1997) that will lead to the same "diffuse likelihood" (as the resulting likelihood is often called). Francke, Koopman, and de Vos (2010) suggest modification to instead compute a marginal likelihood that is equivalent to the diffuse likelihood only under certain conditions. (For ARIMA component models as in (2), these conditions are essentially that the AR operators are constrained to have zeros outside the unit circle and the "differencing operators" $\Delta_j(B)$ do not depend on any unknown model parameters.)

Forecasting and signal extraction estimation (of the ARIMA components $\mu_t^{(j)}$) can be done using the Kalman filter and a suitable smoother, as is discussed by Commandeur *et al.* (2011). Bell (2004) gives matrix expressions for the results produced by such calculations for the case where all scale factors are equal to 1. (See also McElroy 2008 for simplified expressions for the signal extraction results.) For signal extraction computations, **REGCMPNT** uses a fixed point smoother of reduced dimension (Anderson and Moore 1979) to produce $E(\mu_t^{(j)}|\mathbf{y})$ and $\mathrm{Var}(\mu_t^{(j)}|\mathbf{y})$, as well as $E(h_{jt}\mu_t^{(j)}|\mathbf{y})$ and $\mathrm{Var}(h_{jt}\mu_t^{(j)}|\mathbf{y})$, where $\mathbf{y} = (y_1, \ldots, y_n)^\top$.

The generality of the ARIMA component specifications in (2) that are allowed by **REGCMPNT** raises one caution. To allow for this level of generality in the models, **REGCMPNT** makes no checks on whether the model structure is "identified," this term referring to whether all ARMA parameters and variances in the model are estimable. Hotta (1989) gives identifiability conditions for ARIMA component models, but his results do not cover two important cases allowed by **REGCMPNT**: components with scale factors, and components with fixed parameters. To illustrate this issue with a simple example, suppose one specifies a model with two scaled white noise components, $y_t = h_{1t}\zeta_{1t} + h_{2t}\zeta_{2t}$, with ζ_{1t} and ζ_{2t} having variances σ_1^2 and σ_2^2. This model is not identified in the standard (default) case where $h_{1t} = h_{2t} = 1$ for all t, because we could not then estimate both σ_1^2 and σ_2^2. This model is identified, however, if either (i) h_{1t} does not equal h_{2t} for at least one observed time point t, or (ii) either or both of σ_1^2 and σ_2^2 are fixed. Section 5 briefly illustrates a more realistic example of this kind of thing – a model with time-varying trading-day regression coefficients all following random walk models with unknown variances. Such a model is identified only because the resulting ARIMA components have different scale factors (the trading-day regression variables). Because **REGCMPNT** provides no checks on model identifiability, it is incumbent on the user to assure that any model specified to the program is indeed identifiable.

3. Getting started with REGCMPNT

REGCMPNT operates from a DOS command window. We have often named the executable program `regcmpnt.exe`, though it really can be given any name. Here we will assume we have named it `rgc.exe`. Also, assume we have an input file named `ex1.nml`. The extension `.nml` refers to Fortran namelist input, which is discussed below. Assume that both the program and input files are located in the same directory. From within this directory we enter the following command:

```
rgc ex1
```

Note that the `.nml` input file extension is not needed here as `.nml` is the default input file extension. If, however, the input file had a different extension (e.g., if the input file was named `ex1.txt`), then we would need to include the extension in the input filename above (e.g., `rgc ex1.txt`). If the program and input files were in different directories, then path names would need to be added to the executable program filename, or to the input filename, or both, as appropriate. For example, if the executable program file were in the directory `c:\regcmpnt` and the input file were in `c:\examples`, then if we entered the command from a prompt at `c:\regcmpnt` we would type

```
rgc c:\examples\ex1
```

while if we entered the command from a prompt at `c:\examples` we would type

```
c:\regcmpnt\rgc ex1
```

In both cases the output files (discussed below) would be written to the same directory as the input file, that is, to `c:\examples`.

The input file to **REGCMPNT** is an ASCII file containing Fortran namelist specifications. The input namelists function like commands telling **REGCMPNT** what data to use, what analyses to perform, what model specifications to use, and what output to provide. Table 2 summarizes **REGCMPNT**'s namelists and their functions. The use of the namelists is illustrated with the examples of Sections 4, 5, and 6.

Several comments are in order. First, most of the arguments to the namelists have default values that are used if nothing is specified. This includes such things as the seasonal period (default = 1, i.e., a nonseasonal series), whether to print out results for all estimation iterations (default = no), and the maximum lag on residual autocorrelations (default depends on length of series and the seasonal period). Thus, the namelist arguments can mostly be thought of as means of changing the defaults. Of course, some arguments, such as the time series `data` argument in the `series` namelist, do not have defaults.

Second, namelists only need be included in the input file if the corresponding action is desired. Thus, if the series is not to be transformed, the `transform` namelist is omitted. If the model has no regression variables, the `regression` namelist is omitted. If forecasting is not desired the `forecast` namelist is omitted, etc. The minimal input file would include only a `series` namelist (this is the only required namelist), though the only output that would result from such an input file would be a table of the series values.

Third, namelists can usually be in any order in the input file, though we tend to order them as listed in Table 2 to clarify the specifications of the series, model, and analyses. There is one exception. When multiple `arima` namelists are present, the values given to the `cmpntreg` argument of the `regression` namelist will depend on how the `arima` namelists are ordered. (See the example of Section 5.) Fourth, inclusion of an `arima` namelist without an `estimate` namelist will nonetheless force model estimation (with default estimation options).

Output files from the **REGCMPNT** program are given the same filename as the input file (`ex1` for our illustration here), and the main output file is given the extension `.out` (i.e., `ex1.out`). It repeats the model specifications as read by the program and gives the basic model fitting results, with the amount of output controlled by various arguments in the namelists. The main output file also includes the diagnostic checking results (if the `check` namelist is included) and forecast results for the observed series (if the `forecast` namelist is included). Forecast results for the unobserved components in model (1) are output to other files, however. Table 3 below summarizes the full set of **REGCMPNT** output files.

Namelist	Function
`series`	Read in the time series data (from within the namelist or from another file); specify the series starting date, seasonal period, and a series title.
`transform`	Apply a transformation (logarithm, power transformation) to the series, or make other adjustments (e.g., a length-of-month adjustment).
`regression`	Specify variables x_{it} for the regression mean function of the model, such as variables for fixed seasonal effects, trading-day or holiday effects, or user-defined regression variables (data for the latter must be read in).
`arima`	Specify the ARIMA model for one of the components $\mu_t^{(j)}$, including as many `arima` namelists as there are components in the model (m). Also, specify or read in the corresponding scale factors h_{jt} (if not 1 for all t).
`estimate`	Specify various options for model estimation (changing default settings) such as the maximum number of iterations and whether or not to print out the correlation matrix of the estimated model parameters.
`check`	Specify output of various diagnostic checks – residual autocorrelations and partial autocorrelations (and how many lags), and residual histogram.
`forecast`	Perform forecasting (of ARIMA components, $\mu_t^{(j)}$, and of the observed series), and specify related options (e.g., forecast origin, maximum forecast lead).
`smooth`	Perform signal extraction estimation of the ARIMA components, $\mu_t^{(j)}$, (over the time frame of the observed series, or for just a subset of this).

Table 2: **REGCMPNT** input namelists and their functions.

Output file	Contents
`ex1.out`	Main output file containing model specifications and estimation results, diagnostic checking results, and forecast results for the observed series.
`ex1.inn`	Kalman filter innovations (one-step prediction errors), their variances, and corresponding standardized innovations.
`ex1.frc`	Point forecasts of the ARIMA components.
`ex1.frv`	Forecast error variances of the ARIMA components.
`ex1.est`	Signal extraction point estimates of the ARIMA components.
`ex1.var`	Signal extraction error variances of the ARIMA components.

Table 3: **REGCMPNT** output files (with input file `ex1.nml`).

The files in Table 3 other than `ex1.out` are produced only when requested. That is, `ex1.inn` is produced only if requested in the `estimate` namelist (via `prtinn = T`), `ex1.frc` and `ex1.frv` are produced only if a `forecast` namelist is included, and `ex1.est` and `ex1.var` are produced only if a `smooth` namelist is included. The error variances in `ex1.frv` and `ex1.var` account for error in estimating regression parameters, but not for error in estimating the ARMA parameters and the component innovation variances.

4. Modeling Nile riverflow

We now illustrate the use of **REGCMPNT** to fit the local level model (Commandeur *et al.* 2011, Equation 3) to the data on Nile riverflow, as was done by Durbin and Koopman (2001, Chapter 2). The namelist input file (named `nile.nml`) is as follows:

```
&series  title = 'Volume of Nile river at Aswan, 1871-1970'  start = 1871
data =
 1120   1160    963   1210   1160   1160    813   1230   1370   1140
  995    935   1110    994   1020    960   1180    799    958   1140
 1100   1210   1150   1250   1260   1220   1030   1100    774    840
  874    694    940    833    701    916    692   1020   1050    969
  831    726    456    824    702   1120   1100    832    764    821
  768    845    864    862    698    845    744    796   1040    759
  781    865    845    944    984    897    822   1010    771    676
  649    846    812    742    801   1040    860    874    848    890
  744    749    838   1050    918    986    797    923    975    815
 1020    906    901   1170    912    746    919    718    714    740
&end

&arima   order = 0 1 0   var = 1   &end

&arima   order = 0 0 0   var = 1   &end

&estimate   estim = t   prtiter = t   armacorr = t   prtinn = t   &end

&check   acf = t   pacf = t   maxlag = 15   hist = t   &end

&forecast   maxlead = 10   &end

&smooth   estimate = t   &end
```

Notice that all the namelists have the same format:

```
&[namelist name]  arguments = values  &end
```

Thus, the `series` namelist is delimited by `&series` and `&end`, and in between these the `title`, `start`, and `data` arguments are given values. The value assignments are fairly self-explanatory, though a few points are worth noting. For `title`, the value (a phrase used as a label at certain places in the output files) must be enclosed in quotation marks (single or double). For `start`, the value given here is the beginning year, as this is an annual series. For monthly or quarterly series a different format is used for the `start` value, as illustrated in the next section. The `data` argument is used here to specify the time series data, which here are given as integers, though they could equally well be real values. Alternatively, the time series data could be stored in and read from another file, in which case the `data` argument would be omitted, and the `file` argument included instead (illustrated in the next section). Finally, entries in the namelists can be separated by spaces or commas or both.

The values assigned in the other namelists have the following implications:

`&arima` (1st): This namelist specifies the ARIMA model for the first ARIMA component, $\mu_t^{(1)}$. The `order` argument gives the order of the nonseasonal part of the model in the ARIMA(p, d, q) format used by Box and Jenkins (1970). The model specified here is a random walk, i.e., $(1 - B)\mu_t^{(1)} = \zeta_{1t}$. The `var` argument sets the initial value of $\mathrm{Var}(\zeta_{1t})$ equal to 1, but note the qualification to this discussed below. Since the series is annual there is no seasonal part specified for the ARIMA model.

`&arima` (2nd): This namelist specifies an ARIMA$(0, 0, 0)$ model, that is, a white noise model, for the second ARIMA component, $\mu_t^{(2)} = \zeta_{2t}$. The `var` argument sets the initial value of $\mathrm{Var}(\zeta_{2t})$ equal to 1, but again note the qualification to this discussed below.

`&estimate`: The arguments given tell **REGCMPNT** to (*i*) estimate the model (`estim = t`, specifying `estim = f` would result in just the likelihood being evaluated at the specified values of the model parameters), (*ii*) print out results for each of the nonlinear estimation iterations (`prtiter = t`), (*iii*) print out the correlation matrix of the estimated ARMA parameters (`armacorr = t`), and (*iv*) write the file `nile.inn` containing a table with the Kalman filter innovations, their variances, and the resulting standardized innovations (v_t, F_t, and $v_t/\sqrt{F_t}$ – see Commandeur *et al.* 2011, Section 3) from the estimated model.

`&check`: The argument values request output of the autocorrelations and partial autocorrelations of the model residuals through lag 15, as well as a histogram of the standardized residuals.

`&forecast`: Inclusion of this namelist requests forecast results (point forecasts, their error variances, and prediction intervals) for the ARIMA components and the observed series y_t. The argument `maxlead` specifies that the forecasts be computed up through 10 years ahead.

`&smooth`: The lone argument value (`estimate = t`) tells **REGCMPNT** to produce signal extraction estimates and error variances for the two ARIMA components.

The qualification mentioned above about the initial values of innovation variances in the `arima` namelists is as follows. If none of these variances is constrained to a fixed value, then for ML estimation any one of the variances can be "concentrated out of the likelihood" (see Bell 2004). In such cases **REGCMPNT** concentrates out the variance of the first component, and then maximizes the likelihood over the $m - 1$ ratios $\sigma_2^2/\sigma_1^2, \ldots, \sigma_m^2/\sigma_1^2$, and the unknown ARMA and regression parameters. The estimate of σ_1^2 then follows from an analytic formula, and the estimates of $\sigma_2^2, \ldots, \sigma_m^2$ follow from this estimate and the estimates of the corresponding variance ratios. In this case the initial values of the variances specified in the `arima` namelists are converted to initial values of the corresponding variance ratios. Hence, the specification given above results in an initial value of 1.0 for σ_2^2/σ_1^2, though we would have gotten the same result by setting `var = 100` in both `arima` namelists, or indeed by setting `var` equal to any common value in both. It should be noted that initial values must be specified for all innovation variances (i.e., `var` must be set to some value in all `arima` namelists), with the exception that when the likelihood will be concentrated (no fixed variances), if `var` is not set for the first ARIMA component it is given a default value of 1.0 so that, effectively, the `var` specifications in the remaining `arima` namelists are specifying initial values of the variance ratios. Initial values are not required, however, for ARMA parameters (the default initial values are all 0.1), and initial values are not allowed for regression parameters.

(Actually, to be precise, **REGCMPNT** maximizes the concentrated likelihood over the square roots of the variance ratios, $\sigma_2/\sigma_1, \ldots, \sigma_m/\sigma_1$, squaring their estimates after convergence. This enforces the nonnegativity constraint on the variances. The same thing is done directly with unknown variances when the likelihood cannot be concentrated. See Bell (2004) for further discussion.)

With the above input file, **REGCMPNT** produces the following ML estimates of the two model parameters, which are the two component innovation variances, along with their associated asymptotic standard errors:

$$\begin{aligned} \hat{\sigma}_1^2 &= 1472.7 \qquad \text{std. error}(\hat{\sigma}_1^2) = 1347.6 \\ \hat{\sigma}_2^2 &= 15092 \qquad \text{std. error}(\hat{\sigma}_2^2) = 3176.4 \end{aligned}$$

The point estimates agree closely with those reported in Durbin and Koopman (2001, p. 32). The asymptotic variance-covariance matrix of the vector of estimated model parameters (denoted as ψ) is given by the negative inverse Hessian matrix ($-D^{-1}$) of the (not concentrated) log-likelihood $\ell(\psi) = \log L(y|\psi)$ (for $L(y|\psi)$ note Commandeur *et al.* 2011, Section 3, Equation 16):

$$\text{Var}(\hat{\psi}) = -D^{-1} \quad \text{where } D = [d_{ik}] \text{ with } d_{ik} = \partial^2 \ell(\psi)/\partial\psi_i \partial\psi_k|_{\psi=\hat{\psi}} \tag{4}$$

The derivatives are approximated numerically using standard formulas. The square roots of the diagonal elements of $\text{Var}(\hat{\psi})$ provide the standard errors of the parameter estimates. Notice that the standard error of $\hat{\sigma}_1^2$ is relatively large, so that the coefficient of variation, CV, of $\hat{\sigma}_1^2$ is std. $\text{error}(\hat{\sigma}_1^2)/\hat{\sigma}_1^2 = 0.92$. That is, $\hat{\sigma}_1^2$ is rather poorly estimated relative to its magnitude. The CV of $\hat{\sigma}_2^2$ is only $3176.4/15092 = 0.21$, so $\hat{\sigma}_2^2$ is rather better estimated. Also available from $\text{Var}(\hat{\psi})$ is $\text{Corr}(\hat{\sigma}_1^2, \hat{\sigma}_2^2) = -0.62$, showing that there is a rather strong negative dependence between estimates of the two variances. This sort of result is not unusual, reflecting the fact that increasing the variance of one component while decreasing the variance of another will tend to offset somewhat, so that the overall variation may not change very much (though this also depends on the ARIMA structure of the components), and this has less effect on overall model fit than would be the case if one simply increased one variance and kept the other fixed.

Figure 1 shows additional results produced by **REGCMPNT**. (Note that the graphs themselves are not produced by **REGCMPNT**, but rather were done in the R statistical package R Development Core Team 2011.) The first graph in Figure 1 is simply a plot of the observed time series y_t. The second graph also shows the observed series (now plotted as a dotted line) in addition to the signal extraction estimate (solid line) of the trend component ($\mu_t^{(1)}$), and 90 percent confidence interval limits (dashed lines) for the trend. The trend estimates are taken from the file `nile.est` and the standard deviations used to construct the confidence

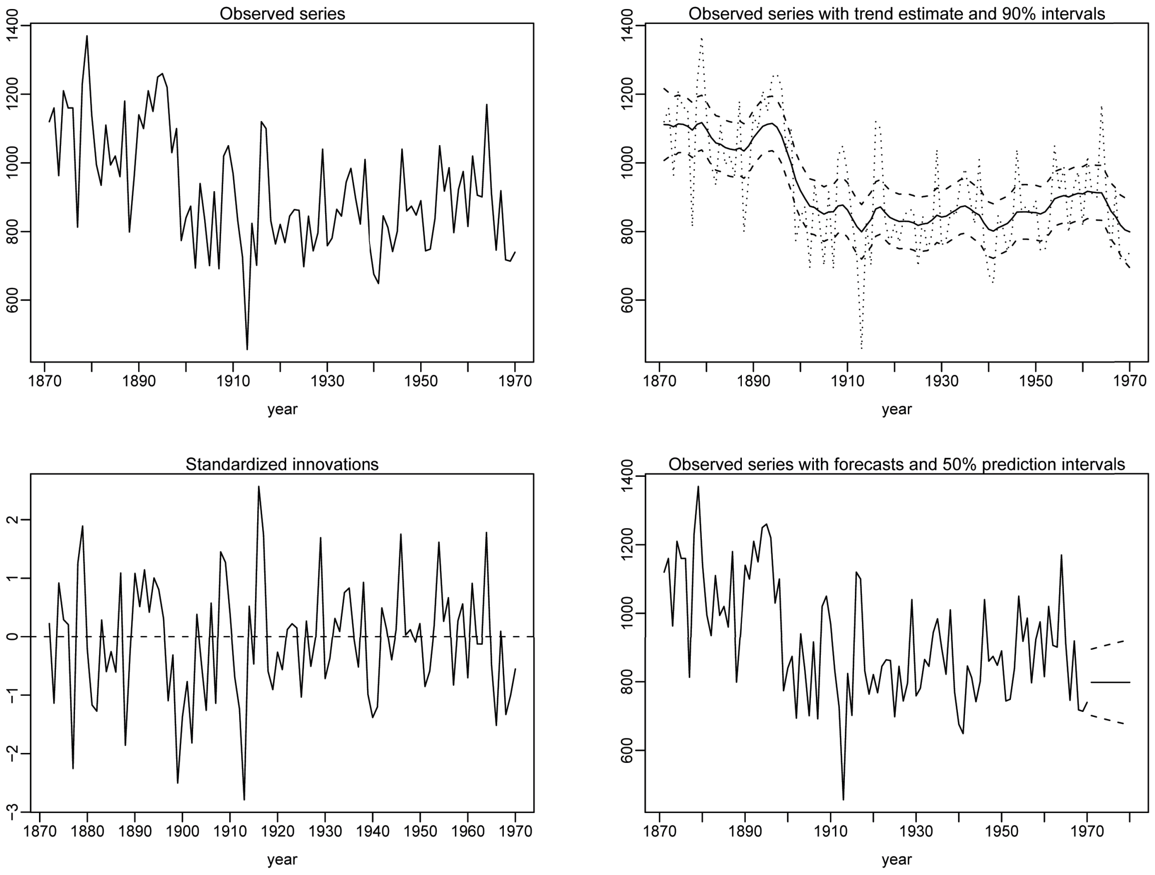

Figure 1: Volume of Nile river at Aswan, 1871–1970.

interval limits are the square roots of the signal extraction variances from the file `nile.var`. The third graph plots the Kalman filter standardized innovations, which are taken from the file `nile.inn`. Finally, the fourth graph again plots the observed series, but also shows point forecasts and 50 percent prediction intervals for y_t for the years 1971,...,1980. The forecast results appear at the end of the main output file (nile.out), though the point forecasts of y_t and the trend component are the same since the second component is white noise (whose forecasts are zero). Also, note that the forecasts of y_t and its trend are constant at the value 798, the constancy being due to the random walk model used for the trend. The results in Figure 1 agree, as near as can be determined from examining graphs, with the corresponding results given in Durbin and Koopman (2001, Chapter 2).

5. A structural model with regression effects

We now illustrate use of **REGCMPNT** to fit a seasonal ARIMA component model that also includes regression terms for trading-day and Easter holiday effects. The time series to which we fit the model is the logarithm of monthly retail sales of U.S. department stores from January, 1967 to December, 1993, a time series that was analyzed with a somewhat different model in Bell (2004), which we discuss later. The regression mean function follows from Bell and Hillmer (1983), and can be written as

$$x_t^\top \beta = \sum_{i=1}^{6} \beta_i T_{it} + \beta_E E_{10,t}. \tag{5}$$

Here $T_{1t}, \ldots, T_{6t}$ are the six trading-day contrast variables defined as (# Mondays in month t) − (# Sundays in month t),…, (# Saturdays in month t) − (# Sundays in month t), and $\beta_1, \ldots, \beta_6$ are the deviations of the Monday, …, Saturday effects from the average daily effect on sales. The corresponding Sunday coefficient is defined implicitly from a constraint that the seven daily effects sum to zero, and so is $\beta_7 = -(\beta_1 + \cdots + \beta_6)$. In addition, to account for length of month effects the series is divided by the length of each month before logs are taken. That is, if s_t is the original series of sales in dollars, we model $y_t = \log(s_t/d_t)$, where d_t is the length in days of month t. The model used for the Easter effect assumes that sales increase by a uniform amount every day before Easter over a window of a given length, which here is 10 days. Thus, $E_{10,t}$ = (# of days in month t falling in the 10-day period before Easter)/10. Actually, **REGCMPNT** also subtracts off the long-term monthly means of the $E_{10,t}$ to remove a fixed seasonal and overall level effect from this term (something also implicit in how the trading-day regressors T_{it} are defined).

The ARIMA component model that we use here is the structural model of Harvey (1989) with a trend component that allows for a random slope, and with a trigonometric seasonal component with a single variance parameter. (Bell 2004, Section 2) notes that these component models have ARIMA representations, so that our overall model can be written as $y_t = x_t^\top \beta + \gamma_t + \mu_t + \epsilon_t$, where ϵ_t is white noise with variance σ_ϵ^2 and

$$(1-B)^2 \mu_t = (1-\theta_\mu B)\zeta_{1t} \tag{6}$$

$$(1+B+\cdots+B^{11})\gamma_t = (1-\theta_1 B - \cdots - \theta_{10} B^{10})\zeta_{2t}. \tag{7}$$

The values of the MA coefficients $\theta_1, \ldots, \theta_{10}$ are fixed at values shown in the namelist input file given below, so the only unknown parameter in (7) is σ_1^2 =Var(ζ_{2t}). The unknown parameters of (6) are θ_μ and σ_1^2 =Var(ζ_{1t}). Harvey formulated this model in a different form, but the only potentially material difference is that Harvey's formulation requires $\theta_\mu \in [0, 1]$, whereas the standard ARIMA constraint on (6) is $\theta_\mu \in [-1, 1]$. In practice, with this model ML estimates of θ_μ are invariably positive, and in fact are often close to or even equal to 1 (Bell and Pugh 1990; Shephard 1993), so this difference in constraints on θ_μ is unlikely to have an effect.

The namelist input file for the model defined by (5)–(7) is as follows:

```
&series   start = 1967,1   period = 12
  title = 'U.S. Retail Sales of Dept Stores, 1/67 - 12/93',
  file = 'c:\retail\bdptrs.dat'
&end

&transform   power = 0.0   &end

&regression   td = t   easter = 10   cmpntreg = 7*2   &end

&arima   order = 0 2 1   macoefs = 0.8   var = 200.   &end
```

```
&arima   order = 0 11 10   diffcoefs = 11*-1.   var = 60.
  macoefs = -0.737378, -0.627978, -0.430368, -0.360770, -0.219736,
            -0.180929, -0.088488, -0.071423, -0.020306, -0.016083
  mafixed = t
&end

&arima   var = 200.   &end

&estimate   estim = t   prtiter = t   maxiter = 100   armacorr = t   &end

&check   acf = t   pacf = f   hist = t   &end

&forecast   maxlead = 24   &end

&smooth   estimate = t   &end
```

Explanation of the entries that differ from those of the namelist file for the Nile riverflow example are as follows:

&series: The start date and title are, of course, appropriate for this series, and since the series is monthly we have `period = 12` and the start date as given in the year, month format. Also, rather than putting the data for the observed series in the `series` namelist it is read from the file `c:\retail\bdptrs.dat` using the `file` argument. Note the input file name is enclosed in quotes.

&transform: The `transform` namelist is included to take logarithms of the series, which results from setting the Box and Cox (1964) power transformation argument to `power = 0.0`.

&regression: The `regression` namelist includes the trading-day regression effects (via `td = t`) and the Easter holiday regression effects with a 10-day window (via `easter = 10`). The `cmpntreg` argument is used to assign individual regression effects ($\beta_i x_{it}$) to the components defined by the `arima` namelists. This matters not for model estimation, but does affect forecasting of the components and signal extraction, as noted by Bell (2005). The values given to `cmpntreg` assign the regression effects to the ARIMA components in the order in which the latter appear in the input file. The notation 7*2 is a Fortran convention that simply repeats the number 2 seven times, and so this is the same as `cmpntreg = 2 2 2 2 2 2 2`. Since the second `arima` namelist is that for the seasonal component in the model, the trading-day and Easter effects are all assigned to the seasonal component.

&arima (1st): This specifies the ARIMA model (6) for the trend component. It sets the initial value of the MA coefficient θ_μ to .8 and the initial value of the variance σ_1^2 to 200, though, as with the previous example, this value will be used only in determining initial values of the variance ratios.

&arima (2nd): This specifies the ARIMA model (7) for the seasonal component. The specification of the "differencing order" as 11, together with the specification of the coefficients of the differencing operator via `diffcoefs = 11*-1`, yields the operator $1 + B + \cdots + B^{11}$. The MA coefficients are set to the values shown, and the argument `mafixed = t` specifies that the coefficients remain fixed at these values. The initial value specified for the variance translates into an initial value of the variance ratio $\sigma_2^2/\sigma_1^2 = 60/200 = 0.3$.

`&arima` (3rd): This specifies the white noise model for the irregular component, with the initial value of the variance leading to a value of 1 for the initial value of $\sigma_\epsilon^2/\sigma_1^2$.

The remaining namelists are similar to those for the Nile riverflow example, with a few different specifications: the `estimate` namelist sets the maximum number of nonlinear estimation iterations to 100 and does not specify printing of the file of standardized innovations; the `check` namelist suppresses printing of the PACF of the residuals and accepts whatever is the default number of lags of residual autocorrelations; and the `forecast` namelist specifies forecasts up to lead 24.

The following shows an excerpt from the main output file of **REGCMPNT** for this example, showing estimation results for the regression part of the model:

```
Regression Model
------------------------------------------------------------------
                            Parameter       Standard
Variable                     Estimate          Error      t-value
------------------------------------------------------------------
Trading Day
  Mon                         -0.0046        0.00209        -2.23
  Tue                          0.0025        0.00210         1.20
  Wed                         -0.0075        0.00206        -3.62
  Thu                          0.0067        0.00209         3.21
  Fri                          0.0079        0.00207         3.81
  Sat                          0.0098        0.00210         4.69
  *Sun (derived)              -0.0149        0.00206        -7.23

Holiday
  Easter 10                    0.0352        0.00420         8.38
------------------------------------------------------------------
Chi-squared Tests
------------------------------------------------------------------
Regression Effect                  df     Chi-square      p-value
------------------------------------------------------------------
Trading Day                         6         244.40         0.00
------------------------------------------------------------------
```

We see that most of the individual trading-day coefficients are easily statistically significant, and jointly (the Wold chi-squared test) they are very highly significant. Since logarithms were taken and the trading-day variables take on the values −1, 0, or 1, 100 times the trading-day parameters can be interpreted as percentages. Hence, the coefficients for Sunday and Saturday imply that sales are depressed about an estimated 1.5 percent when a month has

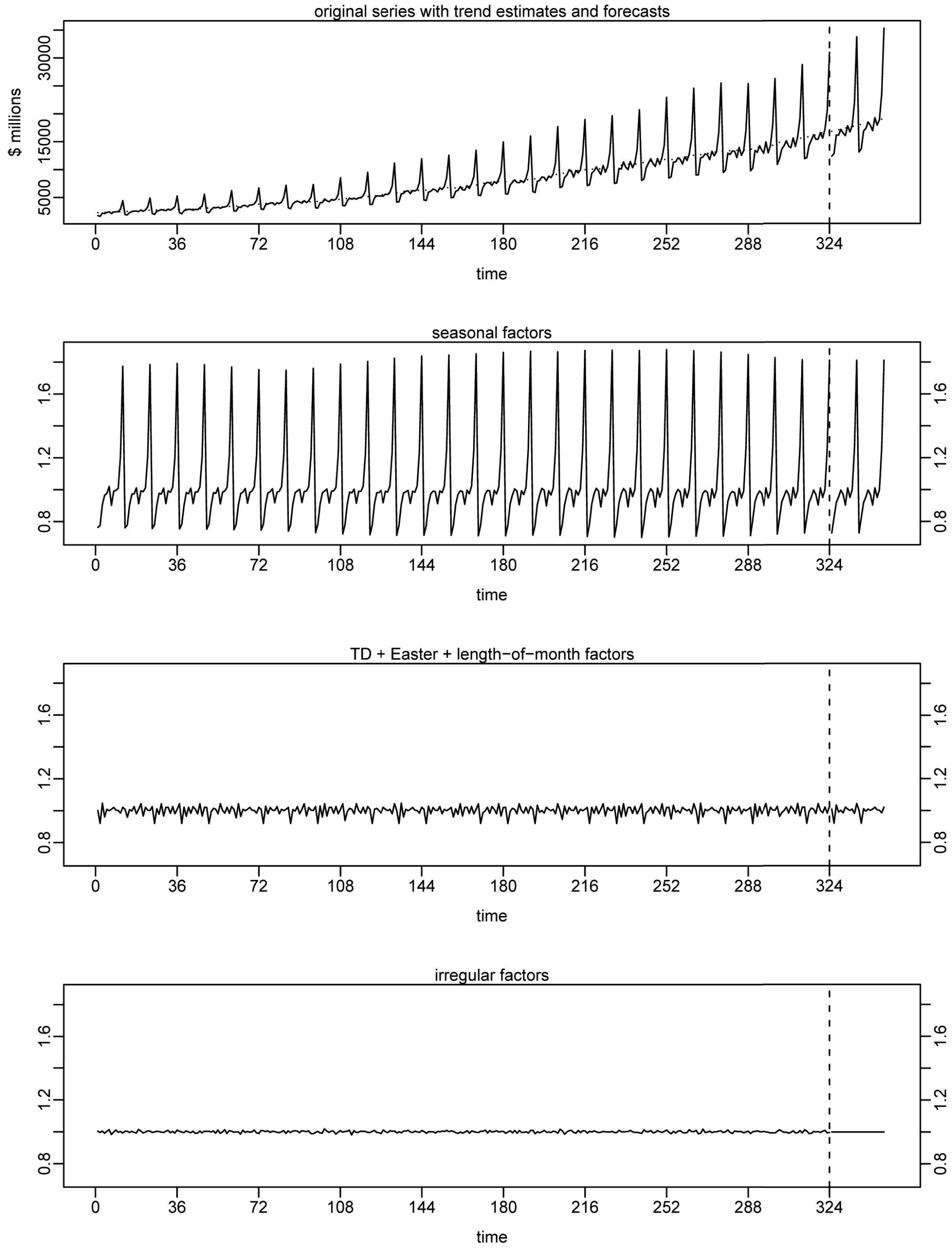

Figure 2: U.S. retail sales of department stores, 1967(1)–1993(12).

five Sundays rather than four, but sales are increased about 1.0 percent by a fifth Saturday. The Easter effect coefficient is also highly significant, and given the definition of the Easter variable it reflects about 3.5 percent of monthly sales coming from the 10 days before Easter that can thus be partially shifted between March and April by variations in the date of Easter.

Model estimation also yielded $\hat{\theta}_\mu = 0.99$, a value sufficiently close to 1.0 that the MA operator in (6) can be cancelled with one of the differences, and the constant term that would be annihilated by the differences added in, yielding the random walk with drift model, $(1-B)\mu_t = \beta_0 + \zeta_{1t}$ (Harvey 1981). Results obtained with this model are very close to those obtained with the model (6).

Figure 2 shows signal extraction estimates and point forecasts of the time series and its components on the original (unlogged) scale. The signal extraction estimates and forecasts in the log scale are simply exponentiated back to the original scale by **REGCMPNT**. The first graph shows the original series with trend estimates superimposed, and corresponding point forecasts of both the series and the trend. The trend estimates and point forecasts appear quite smooth relative to the variation in the original series, with the trend estimates in the middle of each calendar year a little hard to spot as they tend to overlap in the graph with the series values. The remaining three graphs show the signal extraction estimates and point forecasts for the seasonal, regression, and irregular components. The regression component includes the trading-day plus Easter regression effects, and also the length-of-month effects. For these graphs these effects are separated from the seasonal component, which in this plot refers only to the stochastic part of the seasonal. (The **REGCMPNT** output files provide signal extraction and forecasting results both separating the regression and stochastic parts of each component, as well as with the regression and stochastic parts of each component combined.) The vertical scales on all three of these graphs are identical, making it easy to see that, for this series, the seasonal variation is much larger than the variation in the regression and length-of-month effects, which in turn is much larger than the irregular variation.

The results shown above are relevant to the time period of the data (January, 1967 to December, 1993), and results with more recent data could be different. One way results could differ is if trading-day effects changed over time due to changes over time in store hours and shopping patterns of consumers. This motivates consideration of time-varying trading-day parameters, which were considered for this series in Bell (2004), and more generally in Martin and Bell (2004). There an airline model (Box and Jenkins 1970) was used in place of the structural components model, but the main objective was to extend the regression mean function to let the trading-day coefficients follow random walk models. That is, for the trading-day (but not Easter) effects, β_j gets replaced by β_{jt} with $(1 - B)\beta_{jt} = \xi_{jt}$, with the ξ_{jt} independent white noise series. To specify such a model, we use the following `regression` namelist in place of that shown above, and also add the additional `arima` namelists below to the input file:

```
&regression   td = t   easter = 10   cmpntreg = 4 5 6 7 8 9 2   &end
{original three &arima namelists}
&arima   order = 0 1 0   var = 1   tvreg = t   &end
&arima   order = 0 1 0   var = 1   tvreg = t   &end
&arima   order = 0 1 0   var = 1   tvreg = t   &end
&arima   order = 0 1 0   var = 1   tvreg = t   &end
&arima   order = 0 1 0   var = 1   tvreg = t   &end
&arima   order = 0 1 0   var = 1   tvreg = t   &end
```

Note the change in the `cmpntreg` argument to now associate the six trading-day regression variables with the six new `arima` namelists shown, which we assume are included after the three original `arima` namelists for the trend, seasonal, and irregular components. (The Easter effects are still associated with the second ARIMA component, which is the seasonal.) In the new `arima` namelists, the argument specification `tvreg = t` tells **REGCMPNT** that this ARIMA component is a time-varying regression coefficient, which thus must be associated with a regression variable through the `cmpntreg` argument of the `regression` namelist. The initial value of 1.0 given to the variances of the innovations in the random walk models for the trading-day coefficients is a guess at a value that will lead to a reasonable variance ratio, since the first component variance will be concentrated out of the likelihood. So the initial variance ratios for these components are $1/200 = 0.005$. Generally the innovation variances of time-varying trading-day coefficients are expected to be considerably smaller than the innovation variances for the other ARIMA components.

Two additional points are worth noting about the use of the `cmpntreg` argument. First, in assigning regression effects to components, `cmpntreg` takes the regression effects in the order shown in Table 1, which need not match the order in which they are specified in the `regression` namelist. Thus, if the `easter = 10` was put before the `td = t` in the `regression` namelist above, this would not alter the assignments. Second, `cmpntreg` can also assign regression effects to component 0, which is the default. This keeps them as separate effects not assigned to any ARIMA component, and they are then shown separately in the component forecasting and signal extraction results.

The department store sales data analyzed here were produced by the U.S. Census Bureau's monthly retail trade survey. Further information about the survey is available at `http://www.census.gov/retail/`, and more recent data, as well as revised historical data, are available at `http://www.census.gov/retail/mrts/historic_releases.html`. Finally, note that estimates from the survey are generally subject to sampling and nonsampling errors, but in the case of department store sales the sample is sufficiently close to the entire universe of department stores that the sampling error in the data is negligible. This is not the case for the example considered in the next section.

6. Modeling a time series with a sampling error component

The *value of construction put in place (VIP)* is a U.S. Census Bureau publication measuring the value of construction installed or erected at construction sites during a given month. The VIP estimates come from the monthly Construction Progress Reporting Survey (CPRS) augmented with estimates of a non-CPRS component based on regulatory filings, phasing of other Census data, administrative records, and trade association data. Further information, including more recent data and revised historical VIP estimates, can be found at `http://www.census.gov/const/www/c30index.html`.

Nguyen, Bell, and Gomish (2002), hereafter NBG, investigated use of time series modeling and signal extraction methods for improving the VIP estimates. Here, to illustrate how **REGCMPNT** can be applied to model time series with a sampling error component, we show how the general model they developed is applied to the particular VIP series of construction of other educational structures from January, 1997 to December, 2002 (two more years of data than were available for the series analyzed by NBG). NBG developed generalized variance function models for the sampling error relative variances (squares of the sampling error coefficients of variation, or CVs) of the VIP series, to reduce noise in the direct survey relative variance estimates. (The relative variances are, from a Taylor series linearization, approximately the sampling error variances of the log estimates.) The resulting CVs used here mostly range from around 15 to 17 percent over the time frame of the series, reflecting a substantial level of sampling error in the series. NBG also found that AR(2) models fitted the direct estimated sampling error autocorrelations of the VIP series well.

As in NBG we use an airline model (Box and Jenkins 1970) for the signal component Y_t (true series) of the log VIP series y_t, and an AR(2) model for the sampling error e_t. We write the full model as follows:

$$y_t = Y_t + e_t \tag{8}$$

$$(1 - B)(1 - B^{12})Y_t = (1 - \theta_1 B)(1 - \theta_{12} B^{12})\zeta_{1t} \tag{9}$$

$$e_t = h_t \tilde{e}_t \qquad (1 - \phi_1 B - \phi_2 B^2)\tilde{e}_t = \zeta_{2t} \tag{10}$$

where ζ_{1t} and ζ_{2t} are independent white noise series with variances σ_1^2 and σ_2^2, and θ_1 and θ_{12} are parameters satisfying the constraints $|\theta_1| \leq 1$ and $|\theta_{12}| \leq 1$. In (10), h_t is the sampling error CV of $\exp(y_t)$, the (untransformed) VIP survey estimate at time t, and ϕ_1 and ϕ_2 are estimated via the Yule-Walker equations for the AR(2) model (Box and Jenkins 1970, p. 60) using direct estimates of the sampling error autocorrelations at lags 1 and 2 that are averaged over time for the given lag. σ_2^2 is then determined from the formula for the variance of an AR(2) process (Box and Jenkins 1970, p. 62) so that $\text{Var}(\tilde{e}_t) = 1$. The h_t, ϕ_1, ϕ_2, and σ_2^2 are to be held fixed at these values when fitting the model defined by (8)–(10).

The **REGCMPNT** input file for this example is shown below. As with the previous example, the time series data are read from a separate file (here `c:\VIP\nr055.dat`).

```
&series   start = 1997,1   period = 12
  title = 'Other Educational Value of Construction Put-in-Place'
  file = 'c:\VIP\nr055.dat'
&end

&transform   power = 0.0   &end

&arima   order = 0 1 1   sorder = 0 1 1   var = 0.016565   &end

&arima   order = 2 0 0   arcoefs =0.600, 0.246   var = 0.34488   fix = t
h=
 .042  .042  .067  .122  .129  .135  .152  .168  .173  .177  .179  .179
 .179  .182  .179  .177  .175  .170  .162  .165  .152  .149  .149  .135
 .144  .140  .144  .159  .152  .149  .144  .129  .144  .140  .140  .149
 .149  .156  .152  .152  .156  .165  .159  .159  .159  .149  .144  .144
 .152  .159  .156  .156  .162  .156  .165  .162  .165  .168  .170  .159
 .144  .149  .144  .144  .129  .135  .140  .135  .149  .140  .140  .115
&end

&estimate   estim = t   prtiter = t   armacorr = t   &end

&check   acf = t   pacf = t   maxlag = 18   hist = t   &end

&smooth   estimate = t   &end
```

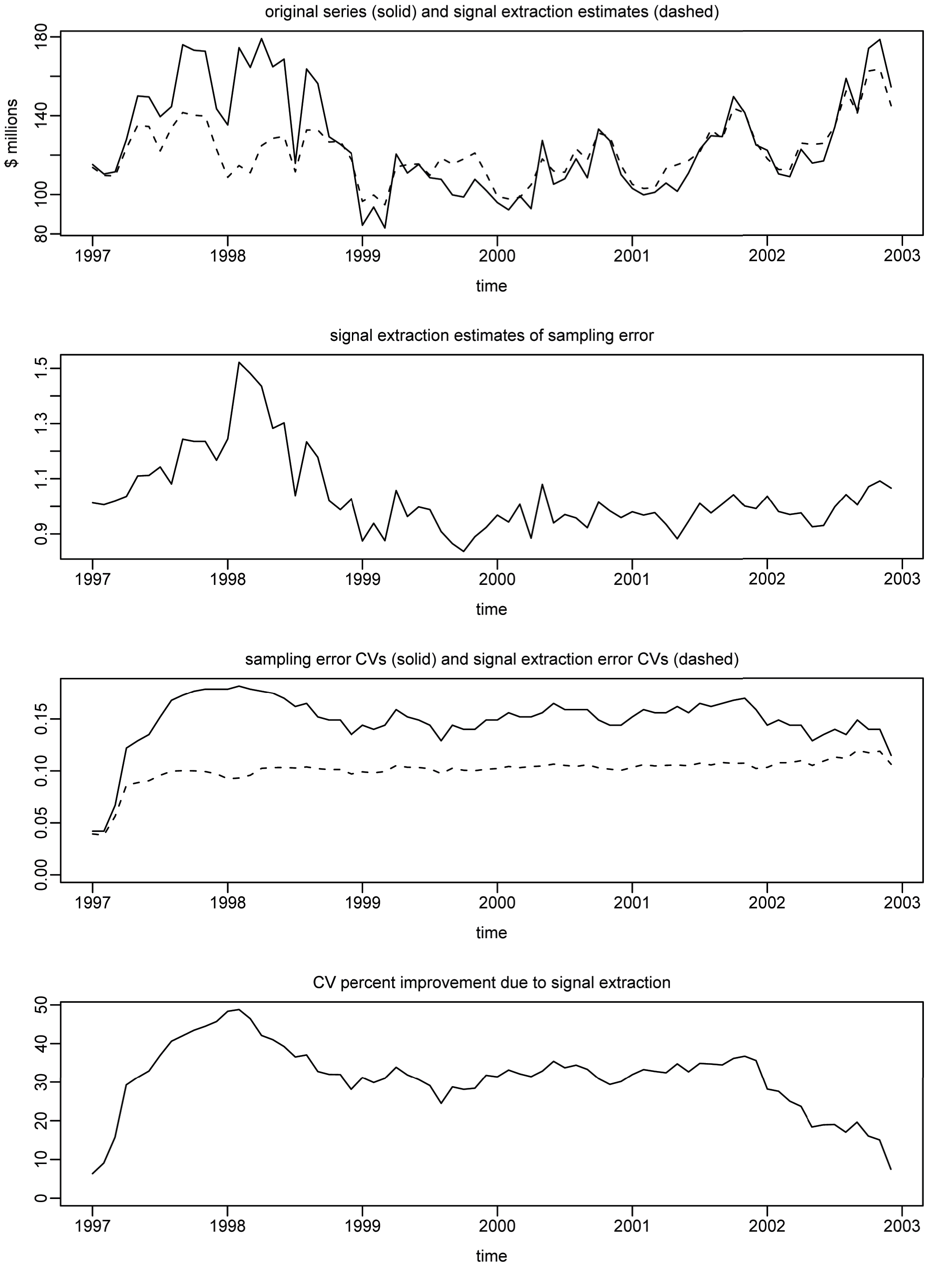

Figure 3: Value of construction put in place, other educational, 1997(1)–2002(12).

The only new features of this input file relative to those of the previous examples involve the two `arima` namelists. The first of these specifies the airline model (9) for Y_t, denoted as an $(0, 1, 1)(0, 1, 1)_{12}$ model in Box and Jenkins (1970, Chapter 9). Note the nonseasonal part of this model is specified with the `order` argument, and the seasonal part with the `sorder` argument. The seasonal period (here 12) is taken from the `period` argument in the `series` namelist. Note also that no initial values are specified for the nonseasonal and seasonal MA parameters θ_1 and θ_{12}; their initial values will thus be set to the ARMA parameter default initial value of .1. The initial value set for σ_1^2 by the `var` argument was taken from a previous analysis of the series. The important point here is that, in contrast to the previous two examples, this initial value is actually for the parameter σ_1^2, and not for a variance ratio. This is because σ_2^2, the innovation variance of the second (sampling error) component, is being held fixed, so that a variance cannot be concentrated out of the likelihood function.

In fact, the argument specification `fix = t` in the second `arima` namelist fixes all the parameters of the model (10) at their specified values: $\phi_1 = 0.600$ and $\phi_2 = 0.246$ in the `arcoefs` argument, and $\sigma_2^2 = 0.34488$ in the `var` argument. (The latter is the value determined so that $\mathrm{Var}(\tilde{e}_t) = 1$ when $\phi_1 = 0.600$ and $\phi_2 = 0.246$.) The other new feature shown in this `arima` namelist is the specification of the scale factors h_t, which are set to the CV values shown via the `h =` argument. As with the input of data values for the series, these values could alternatively be placed in and read from another file by including a `file` argument in the `arima` namelist.

Maximum likelihood parameter estimates for the model defined by (8)–(10) are $\hat{\theta}_1 = 0.47$, $\hat{\theta}_{12} = 0.42$, and $\hat{\sigma}_1^2 = 0.0052$. Signal extraction results from **REGCMPNT** are shown in Figure 3. The first graph plots the original series (on the original scale, i.e., $\exp(y_t)$), along with the signal extraction estimates ($\exp(\hat{Y}_t)$, where $\hat{Y}_t$ is the signal extraction estimate of Y_t). We see that the signal extraction does a large amount of smoothing of the series, which is due to the relatively high level of sampling error in the estimates. The corresponding signal extraction estimates of the sampling error component ($\exp(e_t)$) are plotted in the second graph. These fluctuate around 1.0 (recall that, on the original scale, these are multiplicative factors), with the largest estimated sampling errors occurring in early 1998. The third graph plots the CVs of the sampling error and the signal extraction error, which are approximated by the corresponding standard deviations in the log scale ($[\mathrm{Var}(e_t)]^{.5}$ and $[\mathrm{Var}(Y_t - \hat{Y}_t)]^{.5}$). We see these are lowest at the very beginning of the series, with both rising substantially in early 1997, and with $[\mathrm{Var}(e_t)]^{.5}$ rising further to vary around .15, while $[\mathrm{Var}(Y_t - \hat{Y}_t)]^{.5}$ stabilizes closer to .10. The fourth graph plots the CV percent improvement from signal extraction defined as $100 \times \{[\mathrm{Var}(e_t)]^{.5} - [\mathrm{Var}(Y_t - \hat{Y}_t)]^{.5}\}/[\mathrm{Var}(e_t)]^{.5}$. Corresponding to the results shown in the third graph, the fourth graph shows that the CV improvement is small in the first few and last one or two observations, but is substantial over much of the series. The improvement is often around 30 percent, but reaches close to 50 percent in early 1998 when the sampling CV is the highest. It should be kept in mind, however, that these results on CV improvement from signal extraction are optimistic in that they treat the fitted model as correct.

7. Conclusions

The examples presented illustrate the main capabilities of the **REGCMPNT** program for analyzing RegComponent models. Discussion of the underlying theory and computational approaches was limited, but these topics are covered by Bell (2004). A few features of **REGCMPNT** are worth emphasizing. One is the availability of many regression variables commonly used in modeling seasonal economic time series that are built into the program (through the `regression` namelist). Section 2 discussed these in general terms, and the example of Section 5 illustrated this capability. Another important feature of **REGCMPNT** is the ability to fix parameter values in the ML estimation, including the convenient way one can fix all the parameters of a given ARIMA component. This feature is essential to modeling time series with a sampling error component, and was illustrated in Section 6. Finally, we note the generality of the ARIMA component specifications available in **REGCMPNT**. While the requirement of specifying the component models in ARIMA form eliminates some univariate state space models (the cycle model of Harvey 1989 being one example, as its ARMA representation involves a nonlinear parameter constraint), the models **REGCMPNT** allows are quite general, covering a wide range of commonly used cases. This includes RegARIMA models as are used in **X-12-ARIMA**, the basic structural models of Harvey (1989), models with time-varying regression coefficients, and many others.

Acknowledgments

Thanks are due to all those who have worked over the years to develop the **REGCMPNT** program. Steve Hillmer wrote a Fortran program for signal extraction with ARIMA component models that provided the start towards **REGCMPNT**. Then, Mark Otto, Jim Bozik, Marian Pugh, Larry Bobbitt, and Rich Gagnon all developed important parts of **REGCMPNT**. Brian Monsell has provided assistance with parts of the code shared with the **X-12-ARIMA** program, and Donald Martin has contributed to a recent further extension. But additional thanks are due to Mark Otto, who developed the **REGARIMA** program for analysis of RegARIMA models (which became the modeling part of **X-12-ARIMA**). Mark modified **REGARIMA** to provide the foundation on which **REGCMPNT** was built. This allowed the rest of us to focus on programming only those calculations specific to the state space treatment of RegComponent models, while much of the existing **REGARIMA** code could be used with relatively minor modifications.

Disclaimer

This article is released to inform interested parties of ongoing research and to encourage discussion. The views expressed on statistical, methodological, technical, or operational issues are those of the author and not necessarily those of the U.S. Census Bureau.

References

Anderson B, Moore JB (1979). *Optimal Filtering*. Prentice-Hall, Englewood Cliffs.

Ansley CF, Kohn R (1985). "Estimation, Filtering, and Smoothing in State Space Models with Incompletely Specified Initial Conditions." *The Annals of Statistics*, **13**, 1286–1316.

Bell WR (1984). "Signal Extraction for Nonstationary Time Series." *The Annals of Statistics*, **12**, 646–664.

Bell WR (1991). "Correction: Signal Extraction for Nonstationary Time Series." *The Annals of Statistics*, **19**, 2280.

Bell WR (2004). "On RegComponent Time Series Models and Their Applications." In AC Harvey, SJ Koopman, N Shephard (eds.), *State Space and Unobserved Component Models: Theory and Applications*, chapter 12. Cambridge University Press, Cambridge.

Bell WR (2005). "Some Consideration of Seasonal Adjustment Variances." In *ASA Proceedings of the Joint Statistical Meetings*, pp. 2747–2758. American Statistical Association. URL http://www.census.gov/ts/papers/jsm2005wrb.pdf.

Bell WR, Hillmer SC (1983). "Modeling Time Series with Calendar Variation." *Journal of the American Statistical Association*, **78**, 526–534.

Bell WR, Hillmer SC (1990). "The Time Series Approach to Estimation for Repeated Surveys." *Survey Methodology*, **16**, 195–215.

Bell WR, Hillmer SC (1991). "Initializing the Kalman Filter for Nonstationary Time Series Models." *Journal of Time Series Analysis*, **12**, 283–300.

Bell WR, Pugh MG (1990). "Alternative Approaches to the Analysis of Time Series Components." In AC Singh, P Whitridge (eds.), *Analysis of Data in Time, Proceedings of the 1989 International Symposium*, pp. 105–116. Statistics Canada.

Box GEP, Cox DR (1964). "An Analysis of Transformations." *Journal of the Royal Statistical Society B*, **26**, 211–252.

Box GEP, Jenkins GM (1970). *Time Series Analysis: Forecasting and Control*. San Francisco: Holden Day.

Commandeur JJF, Koopman SJ, Ooms M (2011). "Statistical Software for State Space Methods." *Journal of Statistical Software*, **41**(1), 1–18. URL http://www.jstatsoft.org/v41/i01/.

Durbin J, Koopman SJ (2001). *Time Series Analysis by State Space Methods*. Number 24 in Oxford statistical science series. Oxford University Press, Oxford.

Findley DF, Monsell BC, Bell WR, Otto MC, Chen BC (1998). "New Capabilities and Methods of the **X-12-ARIMA** Seasonal Adjustment Program (with discussion)." *Journal of Business and Economic Statistics*, **16**, 127–177.

Francke MK, Koopman SJ, de Vos AF (2010). "Likelihood Functions for State Space Models with Diffuse Initial Conditions." *Journal of Time Series Analysis*, **31**, 407–414.

Harvey AC (1981). "Finite Sample Prediction and Over-Differencing." *Journal of Time Series Analysis*, **2**, 221–232.

Harvey AC (1989). *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge University Press, Cambridge.

Hotta LK (1989). "Identification of Unobserved Components Models." *Journal of Time Series Analysis*, **10**, 259–270.

Kohn R, Ansley CF (1987). "Signal Extraction for Finite Nonstationary Time Series." *Biometrika*, **74**, 411–421.

Koopman SJ (1997). "Exact Initial Kalman Filtering and Smoothing for Non-Stationary Time Series Models." *Journal of the American Statistical Association*, **92**, 1630–1638.

Martin DEK, Bell WR (2004). "Modeling Time-Varying Trading-Day Effects in Monthly Time Series." In *Proceedings of the American Statistical Association*, pp. 1045–1052. [CD-ROM].

McElroy TS (2008). "Matrix Formulas for Nonstationary ARIMA Signal Extraction." *Econometric Theory*, **24**, 988–1009. Earlier version available at http://www.census.gov/ts/papers/matform3.pdf.

More JJ, Garbow BS, Hillstrom KE (1980). "User Guide for **MINPACK**-1." *Technical report*, Argonne National Laboratory, Argonne, Illinois.

Nguyen TTT, Bell WR, Gomish JM (2002). "Investigating Model-Based Time Series Methods to Improve Estimates from Monthly Value of Construction Put-in-Place Surveys." In *Proceedings of the American Statistical Association*, pp. 2470–2475. [CD-ROM].

R Development Core Team (2011). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

Scott AJ, Smith TMF (1974). "Analysis of Repeated Surveys Using Time Series Methods." *Journal of the American Statistical Association*, **69**, 674–678.

Scott AJ, Smith TMF, Jones RG (1977). "The Application of Time Series Methods to the Analysis of Repeated Surveys." *International Statistical Review*, **45**, 13–28.

Shephard N (1993). "Maximum Likelihood Estimation of Regression Models with Stochastic Trend Components." *Journal of the American Statistical Association*, **88**, 590–595.

US Census Bureau (2009). ***X-12-ARIMA** Reference Manual, Version 0.3.* Time Series Staff, Statistical Research Division, Washington, DC. ISBN 3-900051-07-0, URL http://www.census.gov/srd/www/x12a/.

Wolter KM (1985). *Introduction to Variance Estimation*. Springer-Verlag, New York.

Affiliation:

William R. Bell
Statistical Research Division, room 5K142A
U.S. Census Bureau
4600 Silver Hill Road
Washington, DC 20233, United States of America
E-mail: William.R.Bell@census.gov

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/

Volume 41, Issue 7 *Submitted:* 2010-02-04
May 2011 *Accepted:* 2010-11-26