First off, I should say that this is a rather trivial package, mostly intended for my own use in testing certain aspects of package management in gretl. That said, maybe somebody might find it useful.

1 The Almon lag model

This package estimates an Almon lag or PDL (Polynomial Distributed Lag) model.\(^1\) This model can be a useful way of reducing the number of parameters to be estimated in the distributed lag model given by

\[
y_t = \sum_{i=0}^{p} \beta_i x_{t-i} + \cdots + \epsilon_t
\]

where the ellipsis indicates the possible presence of regressors besides the lags of \(x\).

In the Almon approach we assume that the \(p+1\) coefficients \(\{\beta_i\}\) can be represented by the ordinates of a polynomial \(P(i)\) of degree \(q\), with \(q\) less than \(p\). In that case we can write

\[
\beta_i = P(i) = \sum_{j=0}^{q} \gamma_j i^j \quad i = 0, \ldots, p
\]

The equations (2) may be written out explicitly as

\[
\begin{bmatrix}
\beta_0 \\
\beta_1 \\
\beta_2 \\
\vdots \\
\beta_p \\
\end{bmatrix} =
\begin{bmatrix}
1 & 1 & 1 & \cdots & 1 \\
1 & 2 & 4 & \cdots & 2^q \\
1 & 3 & 9 & \cdots & 3^q \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & p & p^2 & \cdots & p^q \\
\end{bmatrix}
\begin{bmatrix}
\gamma_0 \\
\gamma_1 \\
\gamma_2 \\
\vdots \\
\gamma_q \\
\end{bmatrix}
\]

Substituting from (2) into (1) we get

\[
y_t = \sum_{i=0}^{p} \sum_{j=0}^{q} y_j i^j x_{t-i} + \cdots + \epsilon_t
\]

where \(z_{jt} = \sum_{i=0}^{p} i^j x_{t-i}\). We can think of the \(z_{jt}\)s as “artificial regressors,” which can be constructed quite easily given the time series for \(x\). The procedure is then to run an OLS regression of \(y\) on the \(z_{jt}\)s plus any other regular regressors. With estimates of \(\gamma_0, \ldots, \gamma_q\) in hand, we can use (2) to compute estimates of the \(\beta_i\)s. Standard errors for these estimates can be obtained via the delta method.

\(^1\)I need to acknowledge D. S. G. Pollock, from whose exposition of the Almon lag this section is basically stolen—see [http://www.1e.ac.uk/users/dsgp1/COURSES/TOPICS/Almonlag.pdf](http://www.1e.ac.uk/users/dsgp1/COURSES/TOPICS/Almonlag.pdf).
2 The implementation

3 Command-line use

The function available for command-line and scripting use is named \texttt{almonreg}. It returns a bundle (to be described below) and takes the following arguments:

<table>
<thead>
<tr>
<th>name</th>
<th>type</th>
<th>comment</th>
<th>default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(y)</td>
<td>series</td>
<td>the dependent variable</td>
<td>-</td>
</tr>
<tr>
<td>(x)</td>
<td>series</td>
<td>for PDL treatment</td>
<td>-</td>
</tr>
<tr>
<td>(p)</td>
<td>integer</td>
<td>the maximum lag</td>
<td>4</td>
</tr>
<tr>
<td>(q)</td>
<td>integer</td>
<td>the degree of the polynomial</td>
<td>2</td>
</tr>
<tr>
<td>(c)</td>
<td>boolean</td>
<td>include a constant?</td>
<td>1 (yes)</td>
</tr>
<tr>
<td>(X)</td>
<td>list</td>
<td>additional regressors</td>
<td>null</td>
</tr>
<tr>
<td>quiet</td>
<td>boolean</td>
<td>quiet operation?</td>
<td>0 (no)</td>
</tr>
</tbody>
</table>

We illustrate a call to this function by reference to the sample script included in the package, which replicates the analysis of Almon (1965). This script begins with

```
include almonreg.gfn
open almon.gdt --frompkg=almonreg
```

The data file \texttt{almon.gdt} contains two quarterly time series running from 1952Q1 to 1966Q4, namely \(ce\) (capital expenditures) and \(ca\) (capital appropriations). These are National Industrial Conference Board data, a variant of the data used by Shirley Almon. The next two lines generate a list, \(X\), containing dummy variables representing the first, second and third quarters, to allow for seasonality.

```
genr dummy
list X = dq1 dq2 dq3
```

We then select the lag and polynomial orders and invoke \texttt{almonreg}, saving the result.

```
scalar p = 5 # maximum lag length
scalar q = 2 # order of Almon polynomial
bundle B = almonreg(ce, ca, p, q, 1, X)
```

If we hadn’t wanted to include seasonal dummies the function call could have been abbreviated to

```
bundle B = almonreg(ce, ca, p)
```

in which case the model would have used \(q = 2\) and included a constant, these being the defaults.

4 GUI use

References