1 Purpose

nag_tsa_resid_corr (g13asc) is a diagnostic checking function suitable for use after fitting a Box–Jenkins ARMA model to a univariate time series using nag_tsa_multi_inp_model_estim (g13bec). The residual autocorrelation function is returned along with an estimate of its asymptotic standard errors and correlations. Also, nag_tsa_resid_corr (g13asc) calculates the Box–Ljung portmanteau statistic and its significance level for testing model adequacy.

2 Specification

```c
#include <nag.h>
#include <nagg13.h>

void nag_tsa_resid_corr (Nag_ArimaOrder *arimav, Integer n, const double v[],
    Integer m, const double par[], Integer narma, double r[], double rc[],
    Integer tdrc, double *chi, Integer *df, double *siglev, NagError *fail)
```

3 Description

Consider the univariate multiplicative autoregressive-moving average model

\[ \phi(B)\Phi(B^s)(W_t - \mu) = \theta(B)\Theta(B^s)\epsilon_t \]  

(1)

where \( W_t \), for \( t = 1,2,\ldots,n \), denotes a time series and \( \epsilon_t \), for \( t = 1,2,\ldots,n \), is a residual series assumed to be Normally distributed with zero mean and variance \( \sigma^2 > 0 \). The \( \epsilon_t \)'s are also assumed to be uncorrelated. Here \( \mu \) is the overall mean term, \( s \) is the seasonal period and \( B \) is the backward shift operator such that \( BW_t = W_{t-s} \). The polynomials in (1) are defined as follows:

\[ \phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p \]

is the non-seasonal autoregressive (AR) operator;

\[ \theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \cdots - \theta_q B^q \]

is the non-seasonal moving average (MA) operator;

\[ \Phi(B^s) = 1 - \Phi_1 B^s - \Phi_2 B^{2s} - \cdots - \Phi_p B^{ps} \]

is the seasonal AR operator; and

\[ \Theta(B^s) = 1 - \Theta_1 B^s - \Theta_2 B^{2s} - \cdots - \Theta_Q B^{qs} \]

is the seasonal MA operator. The model (1) is assumed to be stationary, that is the zeros of \( \phi(B) \) and \( \Phi(B^s) \) are assumed to lie outside the unit circle. The model (1) is also assumed to be invertible, that is the zeros of \( \theta(B) \) and \( \Theta(B^s) \) are assumed to lie outside the unit circle. When both \( \Phi(B^s) \) and \( \Theta(B^s) \) are absent from the model, that is when \( P = Q = 0 \), then the model is said to be non-seasonal.

The estimated residual autocorrelation coefficient at lag \( l \), \( \hat{r}_l \), is computed as:

\[ \hat{r}_l = \frac{\sum_{t=l+1}^{n} (\hat{\epsilon}_{t-l} - \bar{\epsilon})(\hat{\epsilon}_t - \bar{\epsilon})}{\sum_{t=1}^{n} (\hat{\epsilon}_t - \bar{\epsilon})^2} \], \quad l = 1,2,\ldots \]

where \( \hat{\epsilon}_t \) denotes an estimate of the \( t \)th residual, \( \epsilon_t \), and \( \bar{\epsilon} = \sum_{t=1}^{n} \hat{\epsilon}_t / n \). A portmanteau statistic, \( Q_{(m)} \), is calculated from the formula (see Box and Ljung (1978)):
where $m$ denotes the number of residual autocorrelations computed. (Advice on the choice of $m$ is given in Section 9.) Under the hypothesis of model adequacy, $Q(m)$ has an asymptotic $\chi^2$ distribution on $m - p - q - P - Q$ degrees of freedom. Let $\hat{r}^T = (\hat{r}_1, \hat{r}_2, \ldots, \hat{r}_m)$ then the variance-covariance matrix of $\hat{r}$ is given by:

$$
\text{Var}(\hat{r}) = \left[ I_m - X(X^TX)^{-1}X^T \right]/n.
$$

The construction of the matrix $X$ is discussed in McLeod (1978). (Note that the mean, $\mu$, and the residual variance, $\sigma^2$, play no part in calculating Var($\hat{r}$) and therefore are not required as input to nag_tsa_resid_corr (g13asc).)

4 References


5 Arguments

1: arimav – Nag_ArimaOrder *

   Pointer to structure of type Nag_ArimaOrder with the following members:

   - p – Integer  \hspace{1cm} \text{Input}
   - d – Integer  \hspace{1cm} \text{Input}
   - q – Integer  \hspace{1cm} \text{Input}
   - bigp – Integer \hspace{1cm} \text{Input}
   - bигd – Integer \hspace{1cm} \text{Input}
   - bigq – Integer \hspace{1cm} \text{Input}
   - s – Integer  \hspace{1cm} \text{Input}

   \text{On entry:} these seven members of \texttt{arimav} must specify the orders vector ($p, d, q, P, D, Q, s$), respectively, of the ARIMA model for the output noise component.

   $p$, $q$, $P$ and $Q$ refer, respectively, to the number of autoregressive ($\phi$), moving average ($\theta$), seasonal autoregressive ($\Phi$) and seasonal moving average ($\Theta$) arguments.

   $d$, $D$ and $s$ refer, respectively, to the order of non-seasonal differencing, the order of seasonal differencing and the seasonal period.

   \text{Constraints:}

   - arimav->p, arimav->q, arimav->bigp, arimav->bigq, arimav->s $\geq 0$,
   - arimav->p + arimav->q + arimav->bigp + arimav->bigq $> 0$,

   if arimav->s = 0, then arimav->bigp = 0 and arimav->bigq = 0.

2: n – Integer  \hspace{1cm} \text{Input}

   \text{On entry:} the number of observations in the residual series, $n$.

   \text{Constraint:} n \geq 3.
3: \( v[n] \) – const double

*Input*

*On entry:* \( v[t-1] \) must contain an estimate of \( \varepsilon_t \), for \( t = 1,2,\ldots,n \).

*Constraint:* \( v \) must contain at least two distinct elements.

4: \( m \) – Integer

*Input*

*On entry:* the value of \( m \), the number of residual autocorrelations to be computed. See Section 9 for advice on the value of \( m \).

*Constraint:* \( n_{arma} < m < n \).

5: \( \text{par}[n_{arma}] \) – const double

*Input*

*On entry:* the parameter estimates in the order \( \phi_1, \phi_2, \ldots, \phi_p, \theta_1, \theta_2, \ldots, \theta_q, \Phi_1, \Phi_2, \ldots, \Phi_P, \Theta_1, \Theta_2, \ldots, \Theta_Q \) only.

*Constraint:* the elements in \( \text{par} \) must satisfy the stationarity and invertibility conditions.

6: \( n_{arma} \) – Integer

*Input*

*On entry:* the number of ARMA arguments, \( \phi, \theta, \Phi \) and \( \Theta \) arguments, i.e., \( n_{arma} = p + q + P + Q \).

*Constraint:* \( n_{arma} = \text{arimav} \rightarrow p + \text{arimav} \rightarrow q + \text{arimav} \rightarrow \text{bigp} + \text{arimav} \rightarrow \text{bigq} \).

7: \( r[m] \) – double

*Output*

*On exit:* an estimate of the residual autocorrelation coefficient at lag \( l \), for \( l = 1,2,\ldots,m \). If \( \text{fail.code} = \text{NE_G13AS_ZERO_VAR} \) on exit then all elements of \( r \) are set to zero.

8: \( \text{rc}[m \times \text{tdrc}] \) – double

*Output*

*On exit:* the estimated standard errors and correlations of the elements in the array \( r \). The correlation between \( r[i-1] \) and \( r[j-1] \) is returned as \( \text{rc}[(i-1) \times \text{tdrc} + j - 1] \) except that if \( i = j \) then \( \text{rc}[(i-1) \times \text{tdrc} + j - 1] \) contains the standard error of \( r[i-1] \). If on exit, \( \text{fail.code} = \text{NE_G13AS_FACT} \) or \( \text{NE_G13AS_DIAG} \), then all off-diagonal elements of \( \text{rc} \) are set to zero and all diagonal elements are set to \( 1/\sqrt{n} \).

9: \( \text{tdrc} \) – Integer

*Input*

*On entry:* the stride separating matrix column elements in the array \( \text{rc} \).

*Constraint:* \( \text{tdrc} \geq m \).

10: \( \text{chi} \) – double *

*Output*

*On exit:* the value of the portmanteau statistic, \( Q_{(m)} \). If \( \text{fail.code} = \text{NE_G13AS_ZERO_VAR} \) on exit then \( \text{chi} \) is returned as zero.

11: \( \text{df} \) – Integer *

*Output*

*On exit:* the number of degrees of freedom of \( \text{chi} \).

12: \( \text{siglev} \) – double *

*Output*

*On exit:* the significance level of \( \text{chi} \) based on \( \text{df} \) degrees of freedom. If \( \text{fail.code} = \text{NE_G13AS_ZERO_VAR} \) on exit then \( \text{siglev} \) is returned as one.

13: \( \text{fail} \) – NagError *

*Input/Output*

The NAG error argument (see Section 3.6 in the Essential Introduction).
6 Error Indicators and Warnings

**NE_2_INT_ARG_LT**

On entry, $\text{tdrc} = \langle \text{value} \rangle$ while $m = \langle \text{value} \rangle$. These arguments must satisfy $\text{tdrc} \geq m$.

**NE_ALLOC_FAIL**

Dynamic memory allocation failed.

**NE_ARIMA_INPUT**

On entry, $\text{arimav} \rightarrow p = \langle \text{value} \rangle$, $\text{arimav} \rightarrow d = \langle \text{value} \rangle$, $\text{arimav} \rightarrow q = \langle \text{value} \rangle$, $\text{arimav} \rightarrow \text{bigp} = \langle \text{value} \rangle$, $\text{arimav} \rightarrow \text{bigd} = \langle \text{value} \rangle$, $\text{arimav} \rightarrow \text{bigq} = \langle \text{value} \rangle$ and $\text{arimav} \rightarrow s = \langle \text{value} \rangle$. Constraints on the members of $\text{arimav}$ are:

- $\text{arimav} \rightarrow p$, $\text{arimav} \rightarrow q$, $\text{arimav} \rightarrow \text{bigp}$, $\text{arimav} \rightarrow \text{bigq}$, $\text{arimav} \rightarrow s \geq 0$,
- $\text{arimav} \rightarrow p + \text{arimav} \rightarrow q + \text{arimav} \rightarrow \text{bigp} + \text{arimav} \rightarrow \text{bigq} > 0$, if $\text{arimav} \rightarrow s = 0$, then $\text{arimav} \rightarrow \text{bigp} = 0$ and $\text{arimav} \rightarrow \text{bigq} = 0$.

**NE_G13AS_AR**

On entry, the autoregressive (or moving average) arguments are extremely close to or outside the stationarity (or invertibility) region. To proceed, you must supply different parameter estimates in the array $\text{par}$.

**NE_G13AS_DIAG**

This is an unlikely exit. At least one of the diagonal elements of $\text{rc}$ was found to be either negative or zero. In this case all off-diagonal elements of $\text{rc}$ are returned as zero and all diagonal elements of $\text{rc}$ set to $1/\sqrt{n}$.

**NE_G13AS_FACT**

On entry, one or more of the AR operators has a factor in common with one or more of the MA operators. To proceed, this common factor must be deleted from the model. In this case, the off-diagonal elements of $\text{rc}$ are returned as zero and the diagonal elements set to $1/\sqrt{n}$. All other output quantities will be correct.

**NE_G13AS_ITER**

This is an unlikely exit brought about by an excessive number of iterations being needed to evaluate the zeros of the AR or MA polynomials. All output arguments are undefined.

**NE_G13AS_ZERO_VAR**

On entry, the residuals are practically identical giving zero (or near zero) variance. In this case $\text{chi}$ is set to zero, $\text{siglev}$ to one and all the elements of $\text{r}$ set to zero.

**NE_INPUT_NARMA**

On entry, $\text{arimav} \rightarrow p = \langle \text{value} \rangle$, $\text{arimav} \rightarrow q = \langle \text{value} \rangle$, $\text{arimav} \rightarrow \text{bigp} = \langle \text{value} \rangle$, $\text{arimav} \rightarrow \text{bigq} = \langle \text{value} \rangle$ while $\text{narma} = \langle \text{value} \rangle$. Constraint: $\text{narma} = \text{arimav} \rightarrow p + \text{arimav} \rightarrow q + \text{arimav} \rightarrow \text{bigp} + \text{arimav} \rightarrow \text{bigq}$.

**NE_INT_3**

On entry, $m = \langle \text{value} \rangle$, $n = \langle \text{value} \rangle$, $\text{narma} = \langle \text{value} \rangle$. Constraint: $\text{narma} < m < n$. 
On entry, \( n = \text{<value>} \).
Constraint: \( n \geq 3 \).

An internal error has occurred in this function. Check the function call and any array sizes. If the call is correct then please contact NAG for assistance.

7 Accuracy
The computations are believed to be stable.

8 Parallelism and Performance
Not applicable.

9 Further Comments

9.1 Timing
The time taken by nag_tsa_resid_corr (g13asc) depends upon the number of residual autocorrelations to be computed, \( m \).

9.2 Choice of \( m \)
The number of residual autocorrelations to be computed, \( m \) should be chosen to ensure that when the ARMA model (1) is written as either an infinite order autoregressive process:

\[
W_t - \mu = \sum_{j=1}^{\infty} \pi_j (W_{t-j} - \mu) + \epsilon_t
\]

or as an infinite order moving average process:

\[
W_t - \mu = \sum_{j=1}^{\infty} \psi_j \epsilon_{t-j} + \epsilon_t
\]

then the two sequences \( \{\pi_1, \pi_2, \ldots\} \) and \( \{\psi_1, \psi_2, \ldots\} \) are such that \( \pi_j \) and \( \psi_j \) are approximately zero for \( j > m \). An overestimate of \( m \) is therefore preferable to an under-estimate of \( m \). In many instances the choice \( m = 10 \) will suffice. In practice, to be on the safe side, you should try setting \( m = 20 \).

9.3 Approximate Standard Errors
When \( \text{fail.code} = \text{NE_G13AS_FACT} \) or \( \text{NE_G13AS_DIAG} \) all the standard errors in \( \text{rc} \) are set to \( 1/\sqrt{n} \). This is the asymptotic standard error of \( \hat{r}_1 \) when all the autoregressive and moving average arguments are assumed to be known rather than estimated.

10 Example
A program to fit an ARIMA(1,1,2) model to a series of 30 observations. 10 residual autocorrelations are computed.
#include <stdio.h>
#include <string.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nagg13.h>

int main(void)
{
    Integer exit_status = 0, i, idf, j, m, *mr = 0, narma, ni, npar;
    Integer nres, nseries, nx;
    NagError fail;
    Nag_ArimaOrder arimav;
    Nag_G13_Opt options;
    Nag_TransfOrder transfv;
    double chi, df, objf, *par = 0, *r = 0, *rc = 0, *res, *sd = 0,
    siglev, *x = 0;

    INIT_FAIL(fail);

    printf("nag_tsa_resid_corr (g13asc) Example Program Results\n\n");

    /* Skip heading in data file */
    #ifdef _WIN32
    scanf_s("%*[\n]");
    #else
    scanf("%*[\n]");
    #endif

    #ifdef _WIN32
    scanf_s("%"NAG_IFMT"%*[\n]", &nx);
    #else
    scanf("%"NAG_IFMT"%*[\n]", &nx);
    #endif
    if (!(x = NAG_ALLOC(nx, double))
        || !(mr = NAG_ALLOC(7, Integer)))
    {
        printf("Allocation failure\n");
        exit_status = -1;
        goto END;
    }

    for (i = 1; i <= nx; ++i)
    {
        #ifdef _WIN32
        scanf_s("%lf", &x[i - 1]);
        #else
        scanf("%lf", &x[i - 1]);
        #endif
        #ifdef _WIN32
        scanf_s("%"NAG_IFMT"%*[\n]");
        #else
        scanf("%"NAG_IFMT"%*[\n]");
        #endif
    }

    #ifdef _WIN32
    scanf_s("%*[\n]");
    #else
    scanf("%*[\n]");
    #endif
    for (i = 1; i <= 7; ++i)
    {
        #ifdef _WIN32
        scanf_s("%"NAG_IFMT"%*[\n]", &mr[i - 1]);
        #else
        scanf("%"NAG_IFMT"%*[\n]", &mr[i - 1]);
        #endif
        #ifdef _WIN32
        scanf_s("%*[\n]");
        #else
        scanf("%*[\n]");
        #endif
    }

    if (!exit_status)
    {
        /* Process the data */
        /* Call nag_tsa_resid_corr (g13asc) */
    }

    return 0;
}

END:
#endif

    if (!(par = NAG_ALLOC(npar, double))
        || !(sd = NAG_ALLOC(npar, double)))
        {
            printf("Allocation failure\n");
            exit_status = -1;
            goto END;
        }
    for (i = 1; i <= npar; ++i)
        par[i - 1] = 0.0;

    nseries = 1;
    arimav.p = mr[0];
    arimav.d = mr[1];
    arimav.q = mr[2];
    arimav.bigp = mr[3];
    arimav.bigd = mr[4];
    arimav.bigq = mr[5];
    arimav.s = mr[6];
    
    /* nag_tsa_options_init (g13bxc).
     * Initialization function for option setting
     */
    nag_tsa_options_init(&options);
    /* nag_tsa_transf_orders (g13byc).
     * Allocates memory to transfer function model orders
     */
    nag_tsa_transf_orders(nseries, &transfv, &fail);
    /* nag_tsa_multi_inp_model_estim (g13bec).
     * Estimation for time series models
     */
    fflush(stdout);
    nag_tsa_multi_inp_model_estim(&arimav, nseries, &transfv, par, npar, nx, x,
        &s, &objf, &df, &options, &fail);

    nres = options.lenres;
    res = options.res;
    if (fail.code != NE_NOERROR)
        {
            printf("Error from nag_tsa_multi_inp_model_estim (g13bec).
            \n%s\n", fail.message);
            exit_status = 1;
            goto END;
        }

    m = 10;
    if (!(r = NAG_ALLOC(m, double))
        || !(rc = NAG_ALLOC(m*m, double)))
        {
            printf("Allocation failure\n");
            exit_status = -1;
            goto END;
        }

    /* nag_tsa_resid_corr (g13asc).
     * Univariate time series, diagnostic checking of residuals,
     * following nag_tsa_multi_inp_model_estim (g13bec)
     */
    nag_tsa_resid_corr(&arimav, nres, res, m, par, narma, r, rc, m, &chi, &sidf, &siglev, &fail);
    if (fail.code != NE_NOERROR)
        {
            printf("Error from nag_tsa_resid_corr (g13asc).\n%s\n", fail.message);
            exit_status = 1;
            goto END;
        }

    printf("RESIDUAL AUTOCORRELATION FUNCTION");
    printf("\n-----------------------------------\n\n");

    g13 – Time Series Analysis
    g13asc

Mark 25

    g13asc.7
for (j = 0; j <= (m-1)/7; j++)
{
    ni = MIN(7, m - j*7);
    printf("LAG K ");
    for (i = 0; i < ni; i++)
        printf("%5"NAG_IFMT" ", i+j*7+1);
    printf("\nR(K) ");
    for (i = 0; i < ni; i++)
        printf("%7.3f", r[i+j*7]);
    printf("\nST.ERROR ");
    for (i = 0; i < ni; i++)
        printf("%7.3f", rc[(m+1)*(i+j*7)]);
    printf("\n---------------------------------------------------------\n");
}

/* nag_tsa_free (g13zc).
 * Freeing function for use with g13 option setting
 */

end:
    NAG_FREE(x);
    NAG_FREE(mr);
    NAG_FREE(par);
    NAG_FREE(sd);
    NAG_FREE(r);
    NAG_FREE(rc);
    return exit_status;

10.2 Program Data

nag_tsa_resid_corr (g13asc) Example Program Data

30: nx, length of the time series
-217 -177 -166 -136 -110 -95 -64 -37
-14 -25 -51 -62 -73 -88 -113 -120
-83 -33 -19 21 17 44 44 78
88 122 126 114 85 64: End of time series
1 1 2 0 0 0 0: mr, orders vector of the model

10.3 Program Results

nag_tsa_resid_corr (g13asc) Example Program Results

Parameters to g13bec

nseries................... 1

criteria................... Nag_Exact
cfixed..................... Nag_FALSE
alpha..................... 1.00e-02
beta...................... 1.00e+01
delta..................... 1.00e+03
gamma..................... 1.00e-07
print_level.............. Nag_Soln
outfile.................. stdout

The number of iterations carried out is 15

The final values of the parameters and their standard deviations are

<table>
<thead>
<tr>
<th>i</th>
<th>para[i]</th>
<th>sd</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-0.094096</td>
<td>0.361543</td>
</tr>
<tr>
<td>2</td>
<td>-0.579152</td>
<td>0.295984</td>
</tr>
<tr>
<td>3</td>
<td>-0.611889</td>
<td>0.182241</td>
</tr>
<tr>
<td>4</td>
<td>9.932425</td>
<td>7.050207</td>
</tr>
</tbody>
</table>

The residual sum of squares = 9.436281e+03
The objective function = 9.762154e+03
The degrees of freedom = 25.00

**RESIDUAL AUTOCORRELATION FUNCTION**

<table>
<thead>
<tr>
<th>LAG</th>
<th>K</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>R(K)</td>
<td></td>
<td>0.030</td>
<td>0.026</td>
<td>-0.039</td>
<td>0.043</td>
<td>-0.129</td>
<td>-0.062</td>
<td>-0.218</td>
</tr>
<tr>
<td>ST.ERROR</td>
<td></td>
<td>0.011</td>
<td>0.116</td>
<td>0.122</td>
<td>0.147</td>
<td>0.171</td>
<td>0.171</td>
<td>0.179</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LAG</th>
<th>K</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>R(K)</td>
<td></td>
<td>-0.105</td>
<td>-0.024</td>
<td>-0.072</td>
</tr>
<tr>
<td>ST.ERROR</td>
<td></td>
<td>0.182</td>
<td>0.182</td>
<td>0.184</td>
</tr>
</tbody>
</table>