NAG Library Function Document

nag_tsa_resid_corr (g13asc)

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

```
#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

$$\phi(B)\Phi(B^s)(W_t - \mu) = \theta(B)\Theta(B^s)\epsilon_t \tag{1}$$

where W_t, for $t = 1, 2, \ldots, n$, denotes a time series and ϵ_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 ϵ_t's are also assumed to be uncorrelated. Here μ is the overall mean term, s is the seasonal period and B is the backward shift operator such that $B^r W_t = W_{t-r}$. 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 tth residual, ϵ_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)):

$$Q_{(m)} = n(n+2)\sum_{l=1}^{m} \hat{r}_l^2/(n-l)$$

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 χ^2 distribution on $m-p-q-P-Q$ degrees of freedom. Let $\hat{r}^{\mathrm{T}} = (\hat{r}_1, \hat{r}_2, \ldots, \hat{r}_m)$ then the variance-covariance matrix of $\hat{r}$ is given by:

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

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

4 References

Box G E P and Ljung G M (1978) On a measure of lack of fit in time series models *Biometrika* **65** 297–303

McLeod A I (1978) On the distribution of the residual autocorrelations in Box–Jenkins models *J. Roy. Statist. Soc. Ser. B* **40** 296–302

5 Arguments

1: **arimav** – Nag_ArimaOrder *

Pointer to structure of type Nag_ArimaOrder with the following members:

p – Integer
d – Integer *Input*
q – Integer *Input*
bigp – Integer *Input*
bigd – Integer *Input*
bigq – Integer *Input*
s – Integer *Input*

On entry: these seven members of **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 (ϕ), moving average (θ), seasonal autoregressive (Φ) and seasonal moving average (Θ) arguments.

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

Constraints:

arimav→p, **arimav→q**, **arimav→bigp**, **arimav→bigq**, **arimav→s** ≥ 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 *Input*

On entry: the number of observations in the residual series, n.

Constraint: **n** ≥ 3.

3: **v**[**n**] – const double *Input*

On entry: **v**$[t-1]$ must contain an estimate of ϵ_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: **narma** $<$ **m** $<$ **n**.

5: **par**[**narma**] – 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 **par** must satisfy the stationarity and invertibility conditions.

6: **narma** – Integer *Input*

On entry: the number of ARMA arguments, ϕ, θ, Φ and Θ arguments, i.e., **narma** $= p + q + P + Q$.

Constraint: **narma** = **arimav**→**p** + **arimav**→**q** + **arimav**→**bigp** + **arimav**→**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 **fail.code** = NE_G13AS_ZERO_VAR on exit then all elements of **r** are set to zero.

8: **rc**[**m** × **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 **rc**$[(i-1) \times$ **tdrc** $+ j - 1]$ except that if $i = j$ then **rc**$[(i-1) \times$ **tdrc** $+ j - 1]$ contains the standard error of **r**$[i-1]$. If on exit, **fail.code** = NE_G13AS_FACT or NE_G13AS_DIAG, then all off-diagonal elements of **rc** are set to zero and all diagonal elements are set to $1/\sqrt{n}$.

9: **tdrc** – Integer *Input*

On entry: the stride separating matrix column elements in the array **rc**.

Constraint: **tdrc** $\geq$ **m**.

10: **chi** – double * *Output*

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

11: **df** – Integer * *Output*

On exit: the number of degrees of freedom of **chi**.

12: **siglev** – double * *Output*

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

13: **fail** – NagError * *Input/Output*

The NAG error argument (see Section 2.7 in How to Use the NAG Library and its Documentation).

6 Error Indicators and Warnings

NE_2_INT_ARG_LT

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

NE_ALLOC_FAIL

Dynamic memory allocation failed.

NE_ARIMA_INPUT

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

arimav→p, **arimav→q**, **arimav→bigp**, **arimav→bigq**, **arimav→s** ≥ 0, **arimav→p** + **arimav→q** + **arimav→bigp** + **arimav→bigq** > 0, if **arimav→s** $= 0$, then **arimav→bigp** $= 0$ and **arimav→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 **par**.

NE_G13AS_DIAG

This is an unlikely exit. At least one of the diagonal elements of **rc** was found to be either negative or zero. In this case all off-diagonal elements of **rc** are returned as zero and all diagonal elements of **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 **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 **chi** is set to zero, **siglev** to one and all the elements of **r** set to zero.

NE_INPUT_NARMA

On entry, **arimav→p** $= \langle value \rangle$, **arimav→q** $= \langle value \rangle$, **arimav→bigp** $= \langle value \rangle$, **arimav→bigq** $= \langle value \rangle$ while **narma** $= \langle value \rangle$.
Constraint: **narma** = **arimav→p** + **arimav→q** + **arimav→bigp** + **arimav→bigq**.

NE_INT_3

On entry, **m** $= \langle value \rangle$, **n** $= \langle value \rangle$, **narma** $= \langle value \rangle$.
Constraint: **narma** $<$ **m** $<$ **n**.

NE_INT_ARG_LT

On entry, **n** = ⟨*value*⟩.
Constraint: **n** ≥ 3.

NE_INTERNAL_ERROR

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

nag_tsa_resid_corr (g13asc) is not threaded in any implementation.

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:

$$W_t - \mu = \sum_{j=1}^{\infty} \pi_j \left(W_{t-j} - \mu \right) + \epsilon_t$$

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 π_j and ψ_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 **fail.code** = NE_G13AS_FACT or NE_G13AS_DIAG all the standard errors in **rc** are set to $1/\sqrt{n}$. This is the asymptotic standard error of $\hat{r}_l$ 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.

10.1 Program Text

```
/* nag_tsa_resid_corr (g13asc) Example Program.
 *
 * NAGPRODCODE Version.
 *
 * Copyright 2016 Numerical Algorithms Group.
 *
 * Mark 26, 2016.
 */

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

```
#else
  scanf("%*[^\n]");
#endif

  npar = mr[0] + mr[2] + mr[3] + mr[5] + 1;
  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,
                                nseries, sd, &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;
  }

  narma = mr[0] + mr[2] + mr[3] + mr[5];
  /* 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, &idf, &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("\nRESIDUAL AUTOCORRELATION FUNCTION");
  printf("\n---------------------------------\n\n");
  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 (g13xzc).
   * Freeing function for use with g13 option setting
   */
  nag_tsa_free(&options);

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

   i            para[i]                sd
   1           -0.094096           0.361543
   2           -0.579152           0.295984
   3           -0.611889           0.182241
   4            9.932425           7.050207


The residual sum of squares =     9.436281e+03

The objective function =     9.762154e+03

The degrees of freedom =    25.00
```

```
RESIDUAL AUTOCORRELATION FUNCTION
---------------------------------

LAG  K      1      2      3      4      5      6      7
R(K)      0.030  0.026 -0.039  0.043 -0.129 -0.062 -0.218
ST.ERROR  0.011  0.116  0.122  0.147  0.171  0.171  0.179
---------------------------------------------------------
LAG  K      8      9     10
R(K)     -0.105 -0.024 -0.072
ST.ERROR  0.182  0.182  0.184
---------------------------------------------------------
```