

NAG Library Function Document

nag_tsa_cross_corr (g13bcc)

1 Purpose

nag_tsa_cross_corr (g13bcc) calculates cross-correlations between two time series.

2 Specification

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

void nag_tsa_cross_corr (const double x[], const double y[], Integer nxy,
                        Integer nl, double *s, double *r0, double r[], double *stat,
                        NagError *fail)
```

3 Description

Given two series x_1, x_2, \dots, x_n and y_1, y_2, \dots, y_n the function calculates the cross-correlations between x_t and lagged values of y_t :

$$r_{xy}(l) = \frac{\sum_{t=1}^{n-l} (x_t - \bar{x})(y_{t+l} - \bar{y})}{ns_x s_y}, \quad l = 0, 1, \dots, L$$

where

$$\bar{x} = \frac{\sum_{t=1}^n x_t}{n}$$

$$s_x^2 = \frac{\sum_{t=1}^n (x_t - \bar{x})^2}{n}$$

and similarly for y .

The ratio of standard deviations s_y/s_x is also returned, and a portmanteau statistic is calculated:

$$\mathbf{stat} = n \sum_{l=1}^L r_{xy}(l)^2.$$

Provided n is large, L much less than n , and both x_t, y_t are samples of series whose true autocorrelation functions are zero, then, under the null hypothesis that the true cross-correlations between the series are zero, **stat** has a χ^2 -distribution with L degrees of freedom. Values of **stat** in the upper tail of this distribution provide evidence against the null hypothesis.

4 References

Box G E P and Jenkins G M (1976) *Time Series Analysis: Forecasting and Control* (Revised Edition) Holden-Day

5 Arguments

- | | | |
|----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|
| 1: | x [nxy] – const double
<i>On entry:</i> the n values of the x series. | <i>Input</i> |
| 2: | y [nxy] – const double
<i>On entry:</i> the n values of the y series. | <i>Input</i> |
| 3: | nxy – Integer
<i>On entry:</i> n , the length of the time series.
<i>Constraint:</i> $\mathbf{nxy} \geq 2$. | <i>Input</i> |
| 4: | nl – Integer
<i>On entry:</i> L , the maximum lag for calculating cross-correlations.
<i>Constraint:</i> $1 \leq \mathbf{nl} < \mathbf{nxy}$. | <i>Input</i> |
| 5: | s – double *
<i>On exit:</i> the ratio of the standard deviation of the y series to the standard deviation of the x series, s_y/s_x . | <i>Output</i> |
| 6: | r0 – double *
<i>On exit:</i> the cross-correlation between the x and y series at lag zero. | <i>Output</i> |
| 7: | r [nl] – double
<i>On exit:</i> r [$l-1$] contains the cross-correlations between the x and y series at lags L , $r_{xy}(l)$, for $l = 1, 2, \dots, L$. | <i>Output</i> |
| 8: | stat – double *
<i>On exit:</i> the statistic for testing for absence of cross-correlation. | <i>Output</i> |
| 9: | fail – NagError *
The NAG error argument (see Section 3.6 in the Essential Introduction). | <i>Input/Output</i> |

6 Error Indicators and Warnings

NE_ALLOC_FAIL

Dynamic memory allocation failed.
See Section 3.2.1.2 in the Essential Introduction for further information.

NE_BAD_PARAM

On entry, argument $\langle value \rangle$ had an illegal value.

NE_INT

On entry, **nl** = $\langle value \rangle$.
Constraint: $\mathbf{nl} \geq 1$.

On entry, **nxy** = $\langle value \rangle$.
Constraint: $\mathbf{nxy} > 1$.

NE_INT_2

On entry, $\mathbf{nl} \geq \mathbf{nxy}$: $\mathbf{nl} = \langle \text{value} \rangle$ and $\mathbf{nxy} = \langle \text{value} \rangle$.

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.

An unexpected error has been triggered by this function. Please contact NAG. See Section 3.6.6 in the Essential Introduction for further information.

NE_NO_LICENCE

Your licence key may have expired or may not have been installed correctly. See Section 3.6.5 in the Essential Introduction for further information.

NE_ZERO_VARIANCE

One or both of the x and y series have zero variance.

7 Accuracy

All computations are believed to be stable.

8 Parallelism and Performance

`nag_tsa_cross_corr` (g13bcc) is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

`nag_tsa_cross_corr` (g13bcc) makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this function. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 Further Comments

If $n < 100$, or $L < 10 \log(n)$ then the autocorrelations are calculated directly and the time taken by `nag_tsa_cross_corr` (g13bcc) is approximately proportional to nL , otherwise the autocorrelations are calculated by utilizing fast Fourier transforms (FFTs) and the time taken is approximately proportional to $n \log(n)$. If FFTs are used then `nag_tsa_cross_corr` (g13bcc) internally allocates approximately $6n$ real elements.

10 Example

This example reads two time series of length 20. It calculates and prints the cross-correlations up to lag 15 for the first series leading the second series and then for the second series leading the first series.

10.1 Program Text

```
/* nag_tsa_cross_corr (g13bcc) Example Program.
 *
 * Copyright 2014 Numerical Algorithms Group.
 *
 * Mark 7, 2002.
 */

#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>
```

```

#include <nagg13.h>

int main(void)
{
    /* Scalars */
    double r0xy, r0yx, statxy, statyx, sxy, syx;
    Integer exit_status, i, nl, nxy;
    NagError fail;

    /* Arrays */
    double *rxy = 0, *ryx = 0, *x = 0, *y = 0;

    INIT_FAIL(fail);

    exit_status = 0;

    printf("nag_tsa_cross_corr (g13bcc) Example Program Results\n");

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

    /* Read series length and number of lags */
#ifdef _WIN32
    scanf_s("%"NAG_IFMT%"NAG_IFMT"%*[\n] ", &nxy, &nl);
#else
    scanf("%"NAG_IFMT%"NAG_IFMT"%*[\n] ", &nxy, &nl);
#endif

    /* Allocate memory */
    if (!(rxy = NAG_ALLOC(nl, double)) ||
        !(ryx = NAG_ALLOC(nl, double)) ||
        !(x = NAG_ALLOC(nxy, double)) ||
        !(y = NAG_ALLOC(nxy, double)))
    {
        printf("Allocation failure\n");
        exit_status = -1;
        goto END;
    }

    /* Read series */
    for (i = 1; i <= nxy; ++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 <= nxy; ++i)
#ifdef _WIN32
        scanf_s("%lf", &y[i-1]);
#else
        scanf("%lf", &y[i-1]);
#endif
#ifdef _WIN32
    scanf_s("%*[\n] ");
#else
    scanf("%*[\n] ");
#endif

    /* Call routine to calculate cross correlations between X and Y */
    /* nag_tsa_cross_corr (g13bcc).
       * Multivariate time series, cross-correlations

```

```

*/
nag_tsa_cross_corr(x, y, nxy, nl, &sxy, &r0xy, rxy, &statxy, &fail);
if (fail.code != NE_NOERROR)
{
    printf("Error from nag_tsa_cross_corr (g13bcc), 1st call.\n%s\n",
           fail.message);
    exit_status = 1;
    goto END;
}

/* Call routine to calculate cross correlations between Y and X */
/* nag_tsa_cross_corr (g13bcc), see above. */
nag_tsa_cross_corr(y, x, nxy, nl, &syx, &r0yx, ryx, &statyx, &fail);
if (fail.code != NE_NOERROR)
{
    printf("Error from nag_tsa_cross_corr (g13bcc), 2nd call.\n%s\n",
           fail.message);
    exit_status = 1;
    goto END;
}

printf("\n");
printf("                Between          Between\n");
printf("                X and Y          Y and X\n");
printf("\n");
printf("Standard deviation ratio%10.4f%15.4f\n", sxy, syx);
printf("\n");
printf("Cross correlation at lag\n");
printf("                0\n");
printf("%10.4f%15.4f\n", r0xy, r0yx);
for (i = 1; i <= nl; ++i)
    printf("                %4"NAG_IFMT"%10.4f%15.4f\n", i, rxy[i-1],
           ryx[i-1]);
printf("\n");
printf("Test statistic                %10.4f%15.4f\n", statxy, statyx);

END:
NAG_FREE(rxy);
NAG_FREE(ryx);
NAG_FREE(x);
NAG_FREE(y);

return exit_status;
}

```

10.2 Program Data

```

nag_tsa_cross_corr (g13bcc) Example Program Data
  20    15
  0.02  0.05  0.08  0.03 -0.05  0.11 -0.01 -0.08 -0.08 -0.11
 -0.18 -0.19 -0.09  0.03  0.10  0.15 -0.14  0.07  0.09  0.16
  3.18  3.21  3.26  3.25  3.08  3.01  3.06  3.17  3.12  3.04
  3.26  3.45  3.33  3.70  3.31  3.81  3.33  2.96  3.28  3.10

```

10.3 Program Results

```

nag_tsa_cross_corr (g13bcc) Example Program Results

                Between          Between
                X and Y          Y and X

Standard deviation ratio    2.0053          0.4987

Cross correlation at lag
  0    0.0568          0.0568
  1    0.0438         -0.0151
  2   -0.3762          0.3955
  3   -0.4864          0.3417
  4   -0.6294          0.5486

```

	5	-0.3871	0.2291
	6	-0.1690	0.3190
	7	-0.0678	0.1980
	8	0.0962	0.0438
	9	0.0788	-0.1428
	10	0.2910	-0.1376
	11	0.0950	-0.0387
	12	0.0547	-0.0380
	13	0.1855	-0.1551
	14	0.0243	-0.1536
	15	0.0034	-0.0696
Test statistic		22.1269	17.2917
