NAG C Library Function Document

nag_tsa_spectrum_univar_cov (g13cac)

1 Purpose

nag_tsa_spectrum_univar_cov (g13cac) calculates the smoothed sample spectrum of a univariate time series using one of four lag windows – rectangular, Bartlett, Tukey or Parzen window.

2 Specification

```c
void nag_tsa_spectrum_univar_cov (Integer nx, Integer mtx, double px, Integer iw, 
Integer mw, Integer ic, Integer nc, double c[], Integer kc, Integer l, 
Nag_LoggedSpectra lg_spect, Integer nxg, double xg[], Integer *ng, 
double stats[], NagError *fail)
```

3 Description

The smoothed sample spectrum is defined as

\[
\hat{f}(\omega) = \frac{1}{2\pi} \left(C_0 + 2 \sum_{k=1}^{M-1} w_k C_k \cos(\omega k)\right),
\]

where \( M \) is the window width, and is calculated for frequency values

\[
\omega_i = \frac{2\pi i}{L}, \quad i = 0, 1, \ldots, \lfloor L/2 \rfloor,
\]

where \( \lfloor \cdot \rfloor \) denotes the integer part.

The autocovariances \( C_k \) may be supplied by the user, or constructed from a time series \( x_1, x_2, \ldots, x_n \), as

\[
C_k = \frac{1}{n-k} \sum_{t=1}^{n-k} x_t x_{t+k},
\]

the fast Fourier transform (FFT) being used to carry out the convolution in this formula.

The time series may be mean or trend corrected (by classical least squares), and tapered before calculation of the covariances, the tapering factors being those of the split cosine bell:

\[
\frac{1}{2}(1 - \cos(\pi(t - \frac{1}{2})/T)), \quad 1 \leq t \leq T
\]

\[
\frac{1}{2}(1 - \cos(\pi(n - t + \frac{1}{2})/T)), \quad n + 1 - T \leq t \leq n
\]

\[
1, \quad \text{otherwise,}
\]

where \( T = \left\lfloor \frac{np}{2} \right\rfloor \) and \( p \) is the tapering proportion.

The smoothing window is defined by

\[
w_k = W\left(\frac{k}{M}\right), \quad k \leq M - 1,
\]

which for the various windows is defined over \( 0 \leq \alpha < 1 \) by rectangular:

\[
W(\alpha) = 1
\]

Bartlett:

\[
W(\alpha) = 1 - \alpha
\]
Tukey:
\[ W(\alpha) = \frac{1}{2}(1 + \cos(\pi\alpha)) \]

Parzen:
\[ W(\alpha) = 1 - 6\alpha^2 + 6\alpha^3, \quad 0 \leq \alpha \leq \frac{1}{2} \]
\[ W(\alpha) = 2(1 - \alpha)^3, \quad \frac{1}{2} < \alpha < 1. \]

The sampling distribution of \( \hat{f}(\omega) \) is approximately that of a scaled \( \chi^2 \) variate, whose degrees of freedom \( d \) is provided by the routine, together with multiplying limits \( mu, ml \) from which approximate 95\% confidence intervals for the true spectrum \( f(\omega) \) may be constructed as \( [ml \times \hat{f}(\omega), \ mu \times \hat{f}(\omega)] \). Alternatively, \( \log f(\omega) \) may be returned, with additive limits.

The bandwidth \( b \) of the corresponding smoothing window in the frequency domain is also provided. Spectrum estimates separated by (angular) frequencies much greater than \( b \) may be assumed to be independent.

4 References

5 Parameters
1: \( nx \) – Integer
   \( On \ entry: \) the length of the time series, \( n \).
   \( Constraint: \ nx \geq 1. \)

2: \( mtx \) – Integer
   \( On \ entry: \) if covariances are to be calculated by the routine (\( ic = 0 \)), \( mtx \) must specify whether the data are to be initially mean or trend corrected.
   \( mtx = 0 \)
   For no correction.
   \( mtx = 1 \)
   For mean correction.
   \( mtx = 2 \)
   For trend correction.
   \( Constraint: \)
   \( if \ ic = 0, 0 \leq mtx \leq 2. \)
   If covariances are supplied (\( ic \neq 0 \)), \( mtx \) is not used.

3: \( px \) – double
   \( On \ entry: \) if covariances are to be calculated by the routine (\( ic = 0 \)), \( px \) must specify the proportion of the data (totalled over both ends) to be initially tapered by the split cosine bell taper. If covariances are supplied (\( ic \neq 0 \)), then \( px \) must specify the proportion of data tapered before the supplied covariances were calculated and after any mean or trend correction. \( px \) is required for the calculation of output statistics. A value of 0.0 implies no tapering.
   \( Constraint: 0.0 \leq px \leq 1.0. \)
4: \textbf{iw} – Integer \hspace{1cm} \textit{Input} \\
\textit{On entry:} the choice of lag window. \textbf{iw} = 1 for rectangular, 2 for Bartlett, 3 for Tukey or 4 for Parzen. \\
\textit{Constraint:} \(1 \leq \textbf{iw} \leq 4\).

5: \textbf{mw} – Integer \hspace{1cm} \textit{Input} \\
\textit{On entry:} the ‘cut-off’ point, \(M\), of the lag window. Windowed covariances at lag \(M\) or greater are zero. \\
\textit{Constraint:} \(1 \leq \textbf{mw} \leq \textbf{nx}\).

6: \textbf{ic} – Integer \hspace{1cm} \textit{Input} \\
\textit{On entry:} indicates whether covariances are to be calculated in the routine or supplied in the call to the routine. \\
\textbf{ic} = 0 \\
\hspace{1cm} \text{Covariances are to be calculated.} \\
\textbf{ic} \neq 0 \\
\hspace{1cm} \text{Covariances are to be supplied.}

7: \textbf{nc} – Integer \hspace{1cm} \textit{Input} \\
\textit{On entry:} the number of covariances to be calculated in the routine or supplied in the call to the routine. \\
\textit{Constraint:} \(\textbf{mw} \leq \textbf{nc} \leq \textbf{nx}\).

8: \textbf{c[nc]} – double \hspace{1cm} \textit{Input/Output} \\
\textit{On entry:} if \textbf{ic} \neq 0, then \textbf{c} must contain the \textbf{nc} covariances for lags from 0 to \((\textbf{nc} - 1)\), otherwise \textbf{c} need not be set. \\
\textit{On exit:} if \textbf{ic} = 0, \textbf{c} will contain the \textbf{nc} calculated covariances. \\
\text{If \textbf{ic} \neq 0, the contents of \textbf{c} will be unchanged.}

9: \textbf{kc} – Integer \hspace{1cm} \textit{Input} \\
\textit{On entry:} if \textbf{ic} = 0, \textbf{kc} must specify the order of the fast Fourier transform (FFT) used to calculate the covariances. \textbf{kc} should be a product of small primes such as \(2^m\) where \(m\) is the smallest integer such that \(2^m \geq \textbf{nx} + \textbf{nc}\), provided \(m \leq 20\). \\
\text{If \textbf{ic} \neq 0, that is covariances are supplied, then \textbf{kc} is not used.} \\
\textit{Constraint:} \(\textbf{kc} \geq \textbf{nx} + \textbf{nc}\). The largest prime factor of \textbf{kc} must not exceed 19, and the total number of prime factors of \textbf{kc}, counting repetitions, must not exceed 20. These two restrictions are imposed by nag_fft_real (c06eac) which performs the FFT.

10: \textbf{l} – Integer \hspace{1cm} \textit{Input} \\
\textit{On entry:} the frequency division, \(L\), of the spectral estimates as \(\frac{2}{L}\). Therefore it is also the order of the FFT used to construct the sample spectrum from the covariances. \textbf{l} should be a product of small primes such as \(2^m\) where \(m\) is the smallest integer such that \(2^m \geq 2M - 1\), provided \(m \leq 20\). \\
\textit{Constraint:} \(1 \geq 2 \times \text{mw} - 1\). The largest prime factor of \textbf{l} must not exceed 19, and the total number of prime factors of \textbf{l}, counting repetitions, must not exceed 20. These two restrictions are imposed by nag_fft_real (c06eac) which performs the FFT.

11: \textbf{lg_spect} – Nag_LoggedSpectra \hspace{1cm} \textit{Input} \\
\textit{On entry:} indicates whether unlogged or logged spectral estimates and confidence limits are required.

\[\text{NP3645/7}\]
lg_spect = Nag_Unlogged
Unlogged.

lg_spect = Nag_Locked
Logged.

Constraint: lg_spect = Nag_Unlogged or Nag_Locked.

12: nxg – Integer

On entry: the dimension of the array xg as declared in the function from which
nag_tsa_spectrum_univar_cov (g13cac) is called.

Constraints:
if ic = 0, nxg ≥ max(kc, 1);
if ic ≠ 0, nxg ≥ 1.

13: xg[nxg] – double

On entry: if the covariances are to be calculated, then xg must contain the nx data points. If
covariances are supplied, xg may contain any values.

On exit: contains the ng spectral estimates, \( \hat{f}(\omega_i) \), for \( i = 0, 1, \ldots, [L/2] \) in xg[0] to xg[ng − 1] respectively (logged if lg_spect = Nag_Locked). The elements xg[i − 1], for \( i = ng + 1, \ldots, nxg \) contain 0.0.

14: ng – Integer *

On exit: the number of spectral estimates, \([L/2] + 1\), in xg.


On exit: four associated statistics. These are the degrees of freedom in stats[0], the lower and upper
95% confidence limit factors in stats[1] and stats[2] respectively (logged if lg_spect = Nag_Locked), and the bandwidth in stats[3].

16: fail – NagError *

The NAG error parameter (see the Essential Introduction).

6 Error Indicators and Warnings

NE_INT
On entry, mw = ⟨value⟩.
Constraint: mw ≥ 1.

On entry, iw is not equal to 1, 2, 3 or 4: iw = ⟨value⟩.

On entry, ic = 0 and mtx > 2: mtx = ⟨value⟩.

On entry, ic = 0 and mtx < 0: mtx = ⟨value⟩.

On entry, nx = ⟨value⟩.
Constraint: nx ≥ 1.

NE_INT_2
On entry, mw > nx: mw = ⟨value⟩, nx = ⟨value⟩.
On entry, l < 2 × mw − 1: l = ⟨value⟩, mw = ⟨value⟩.

On entry, ic is not equal to 0 and nxg < l: nxg = ⟨value⟩, l = ⟨value⟩.

On entry, nc > nx: nc = ⟨value⟩, nx = ⟨value⟩.
On entry, \( nc < mw \): \( nc = \langle value \rangle \), \( mw = \langle value \rangle \).

**NE_INT_3**

On entry, \( iec = 0 \) and \( kc < (nx + nc) \): \( kc = \langle value \rangle \), \( nx = \langle value \rangle \), \( nc = \langle value \rangle \).

On entry, \( iec = 0 \) and \( nxg < \text{max}(kc,l) \): \( nxg = \langle value \rangle \), \( kc = \langle value \rangle \), \( l = \langle value \rangle \).

**NE_CONFID_LIMITS**

The calculation of confidence limit factors has failed.

**NE_PRIME_FACTOR**

\( l \) has a prime factor exceeding 19, or more than 20 prime factors (counting repetitions): \( l = \langle value \rangle \).

\( kc \) has a prime factor exceeding 19, or more than 20 prime factors (counting repetitions): \( kc = \langle value \rangle \).

**NE_REAL**

On entry, \( px = \langle value \rangle \).

Constraint: \( px \leq 1.0 \).

On entry, \( px = \langle value \rangle \).

Constraint: \( px \geq 0.0 \).

**NE_SPECTRAL_ESTIMATES**

One or more spectral estimates are zero. Consult the values in \( xg \) and \( stats \).

**NE_BAD_PARAM**

On entry, parameter \( \langle value \rangle \) had an illegal value.

**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 consult NAG for assistance.

### 7 Accuracy

The FFT is a numerically stable process, and any errors introduced during the computation will normally be insignificant compared with uncertainty in the data.

### 8 Further Comments

\texttt{nag\_tsa\_spectrum\_univar\_cov} (g13cac) carries out two FFTs of length \( kc \) to calculate the covariances and one FFT of length \( l \) to calculate the sample spectrum. The time taken by the routine for an FFT of length \( n \) is approximately proportional to \( n \log(n) \) (see Section 8 of the document for \texttt{nag\_fft\_real} (c06eac) for further details).

### 9 Example

The example program reads a time series of length 256. It selects the mean correction option, a tapering proportion of 0.1, the Parzen smoothing window and a cut-off point for the window at lag 100. It chooses to have 100 auto-covariances calculated and unlogged spectral estimates at a frequency division of \( 2\pi/200 \). It then calls \texttt{nag\_tsa\_spectrum\_univar\_cov} (g13cac) to calculate the univariate spectrum and statistics and prints the autocovariances and the spectrum together with its 95% confidence multiplying limits.
9.1 Program Text

#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nagg13.h>

int main(void)
{
    /* Scalars */
    double px;
    Integer exit_status, i, ic, iw, kc, lf, lg, mtx, mw, nc, ng, nx, nxg;
    NagError fail;
    /* Arrays */
    double *c = 0, *xg = 0;
    double stats[4];

    INIT_FAIL(fail);
    exit_status = 0;
    Vprintf("g13cac Example Program Results\n");
    /* Skip heading in data file */
    Vscanf("%*[^
"]");
    Vscanf("%ld%ld%*[^
"]", &nx, &nc);

    if (nx > 0 && nc > 0)
    {
        mtx = 1;
        px = 0.1;
        iw = 4;
        mw = 100;
        ic = 0;
        kc = 360;
        lf = 200;
        lg = 0;

        if (ic == 0)
        {
            nxg = MAX(kc, lf);
        }
        else
        {
            nxg = lf;
        }

        /* Allocate memory */
        if ( !(c = NAG_ALLOC(nc, double)) ||
            !(xg = NAG_ALLOC(nxg, double)) )
        {
            Vprintf("Allocation failure\n");
            exit_status = -1;
            goto END;
        }

        for (i = 1; i <= nx; ++i)
        {
            Vscanf("%lf", &xg[i-1]);
            Vscanf("%*[^
"]");
        }

        g13cac(nx, mtx, px, iw, mw, ic, nc, c, kc, lf, Nag_Unlogged,
               nxg, xg, &ng, stats, &fail);
        if (fail.code != NE_NOERROR)
        {
            Vprintf("Error from g13cac.\n", fail.message);
            exit_status = 1;
            goto END;
        }
    }
}

END:


Vprintf("\n");
Vprintf("Covariances\n");
for (i = 1; i <= nc; ++i)
{
    Vprintf("%11.4f", c[i-1]);
    if (i % 6 == 0 || i == nc)
        Vprintf("\n");
}
Vprintf("\n");
Vprintf("Degrees of freedom =%4.1f Bandwidth =%7.4f\n", stats[0], stats[3]);
Vprintf("\n");
Vprintf("95 percent confidence limits - Lower =%7.4f "
        "Upper =%7.4f\n", stats[1], stats[2]);
Vprintf("\n");
Vprintf(" Spectrum Spectrum Spectrum" 
        " Spectrum\n");
Vprintf(" estimate estimate estimate" 
        " estimate\n");
for (i = 1; i <= ng; ++i)
{
    Vprintf("%4ld%10.4f", i, xg[i-1]);
    if (i % 4 == 0 || i == ng)
        Vprintf("\n");
}
}

END:
if (c) NAG_FREE(c);
if (xg) NAG_FREE(xg);
return exit_status;

9.2 Program Data

9.2.1 Program Data

5.0 11.0 16.0 23.0 36.0 58.0 29.0 20.0 10.0 8.0 3.0 0.0
5.0 11.0 16.0 23.0 58.0 29.0 20.0 10.0 8.0 3.0 0.0
21.0 40.0 78.0 122.0 103.0 73.0 47.0 35.0 11.0 5.0 16.0 34.0
70.0 81.0 111.0 101.0 73.0 40.0 20.0 16.0 5.0 11.0 22.0 40.0
60.0 80.9 83.4 47.7 47.8 30.7 12.2 9.6 10.2 32.4 47.6 54.0
62.9 85.9 61.2 45.1 36.4 20.9 11.4 37.8 69.8 106.1 100.8 81.6
66.5 34.8 30.6 7.0 19.8 92.5 154.4 125.9 84.8 68.1 38.5 22.8
10.2 24.1 82.9 132.0 130.9 118.1 89.9 66.6 60.0 46.9 41.0 21.3
16.0 6.4 4.1 6.8 14.5 34.0 45.0 43.1 47.5 42.2 28.1 10.1
8.1 2.5 0.0 1.4 5.0 12.2 13.9 35.4 45.8 41.1 30.1 23.9
15.6 6.6 4.0 1.8 8.5 16.6 36.3 49.6 64.2 67.0 70.9 47.8
27.5 8.9 13.2 56.9 121.5 138.3 103.2 85.7 64.6 36.7 24.2 10.7
15.0 40.1 61.5 98.5 124.7 96.3 66.6 64.5 54.1 39.0 20.6 6.7
4.3 22.7 54.8 93.8 95.8 77.2 59.1 44.0 47.0 30.5 16.3 7.3
37.6 74.0 139.0 111.2 101.6 66.2 44.7 17.0 11.3 12.4 3.4 6.0
32.3 54.3 59.7 63.7 63.5 52.2 25.4 13.1 6.8 6.3 7.1 35.6
73.0 85.1 78.0 64.0 41.8 26.2 26.7 12.1 9.5 2.7 5.0 24.4
42.0 63.5 53.8 62.0 48.5 43.9 18.6 5.7 3.6 1.4 9.6 47.4
57.1 103.9 80.6 63.6 37.6 26.1 14.2 5.8 16.7 44.3 63.9 69.0
77.8 64.5 35.7 21.2 11.1 5.7 8.7 36.1 79.7 114.4 109.6 88.8
67.8 47.5 30.6 16.3 9.6 33.2 92.6 151.6 136.3 134.7 83.9 69.4
31.5 13.9 4.4 38.0


### 9.3 Program Results

#### g13cac Example Program Results

**Covariances**

<table>
<thead>
<tr>
<th></th>
<th>1152.9733</th>
<th>937.3289</th>
<th>494.9243</th>
<th>14.8648</th>
<th>-342.8548</th>
<th>-514.6479</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-469.2733</td>
<td>-236.6896</td>
<td>109.0608</td>
<td>441.3498</td>
<td>637.4571</td>
<td>641.9954</td>
</tr>
<tr>
<td>-241.1943</td>
<td>-55.6140</td>
<td>129.4067</td>
<td>267.4248</td>
<td>317.8293</td>
<td>230.2807</td>
<td></td>
</tr>
<tr>
<td>56.4402</td>
<td>-146.4689</td>
<td>-320.9948</td>
<td>-406.4077</td>
<td>-375.6384</td>
<td>-273.5936</td>
<td></td>
</tr>
<tr>
<td>-132.6214</td>
<td>11.0791</td>
<td>126.4843</td>
<td>171.3391</td>
<td>122.6284</td>
<td>-11.5482</td>
<td></td>
</tr>
<tr>
<td>-3.4126</td>
<td>73.2521</td>
<td>98.0831</td>
<td>71.8949</td>
<td>17.0985</td>
<td>-27.5632</td>
<td></td>
</tr>
<tr>
<td>-76.7900</td>
<td>-110.5354</td>
<td>-126.1383</td>
<td>-121.1043</td>
<td>-103.9362</td>
<td>-67.4619</td>
<td></td>
</tr>
<tr>
<td>-10.8678</td>
<td>58.5009</td>
<td>116.4587</td>
<td>140.0961</td>
<td>129.5928</td>
<td>66.3211</td>
<td></td>
</tr>
<tr>
<td>-35.5487</td>
<td>-135.3894</td>
<td>-203.7149</td>
<td>-216.2161</td>
<td>-152.7723</td>
<td>-30.4361</td>
<td></td>
</tr>
<tr>
<td>99.3397</td>
<td>188.9594</td>
<td>204.9047</td>
<td>148.4056</td>
<td>34.4975</td>
<td>-103.7840</td>
<td></td>
</tr>
<tr>
<td>-208.5982</td>
<td>-252.4128</td>
<td>-223.7600</td>
<td>-120.8640</td>
<td>23.3565</td>
<td>156.0956</td>
<td></td>
</tr>
<tr>
<td>227.7642</td>
<td>228.5123</td>
<td>172.3820</td>
<td>87.4911</td>
<td>-21.2170</td>
<td>-117.5282</td>
<td></td>
</tr>
<tr>
<td>-176.3634</td>
<td>-165.1218</td>
<td>-75.1308</td>
<td>67.1634</td>
<td>195.7290</td>
<td>279.3039</td>
<td></td>
</tr>
<tr>
<td>290.8258</td>
<td>225.3811</td>
<td>104.0784</td>
<td>-44.4731</td>
<td>-162.7355</td>
<td>-207.7480</td>
<td></td>
</tr>
<tr>
<td>-165.2444</td>
<td>-48.5473</td>
<td>118.8872</td>
<td>265.0045</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Degrees of freedom = 9.0**  
**Bandwidth = 0.1165**

**95 percent confidence limits -**  
**Lower = 0.4731**  
**Upper = 3.3329**

<table>
<thead>
<tr>
<th>Spectrum</th>
<th>Spectrum</th>
<th>Spectrum</th>
<th>Spectrum</th>
</tr>
</thead>
<tbody>
<tr>
<td>estimate</td>
<td>estimate</td>
<td>estimate</td>
<td>estimate</td>
</tr>
<tr>
<td>1 210.4696</td>
<td>2 428.2020</td>
<td>3 810.1419</td>
<td>4 922.5900</td>
</tr>
<tr>
<td>5 706.1605</td>
<td>6 393.4052</td>
<td>7 207.6481</td>
<td>8 179.0657</td>
</tr>
<tr>
<td>9 170.1320</td>
<td>10 133.0442</td>
<td>11 103.6752</td>
<td>12 103.0644</td>
</tr>
<tr>
<td>13 141.5173</td>
<td>14 194.3041</td>
<td>15 266.5730</td>
<td>16 437.0181</td>
</tr>
<tr>
<td>17 985.3130</td>
<td>18 2023.1574</td>
<td>19 2681.8980</td>
<td>20 2363.7439</td>
</tr>
<tr>
<td>21 1669.9001</td>
<td>22 1012.1320</td>
<td>23 561.4822</td>
<td>24 467.2741</td>
</tr>
<tr>
<td>25 441.9977</td>
<td>26 300.1985</td>
<td>27 172.0184</td>
<td>28 114.7823</td>
</tr>
<tr>
<td>29 79.1533</td>
<td>30 49.4882</td>
<td>31 27.0902</td>
<td>32 16.8081</td>
</tr>
<tr>
<td>33 27.5111</td>
<td>34 59.4429</td>
<td>35 97.0145</td>
<td>36 119.3664</td>
</tr>
<tr>
<td>37 116.6737</td>
<td>38 87.3142</td>
<td>39 54.9570</td>
<td>40 42.9781</td>
</tr>
<tr>
<td>41 46.6097</td>
<td>42 53.6206</td>
<td>43 50.6050</td>
<td>44 36.7780</td>
</tr>
<tr>
<td>45 25.6285</td>
<td>46 24.8555</td>
<td>47 30.2626</td>
<td>48 31.5642</td>
</tr>
<tr>
<td>49 27.3351</td>
<td>50 22.4443</td>
<td>51 18.5418</td>
<td>52 15.2425</td>
</tr>
<tr>
<td>53 12.0207</td>
<td>54 12.6846</td>
<td>55 18.3975</td>
<td>56 19.3058</td>
</tr>
<tr>
<td>57 12.6103</td>
<td>58 7.9511</td>
<td>59 7.1333</td>
<td>60 5.4996</td>
</tr>
<tr>
<td>61 3.4182</td>
<td>62 3.2359</td>
<td>63 5.3836</td>
<td>64 8.5225</td>
</tr>
<tr>
<td>65 10.0610</td>
<td>66 7.9483</td>
<td>67 4.2261</td>
<td>68 3.2631</td>
</tr>
<tr>
<td>69 5.5751</td>
<td>70 7.8491</td>
<td>71 9.3694</td>
<td>72 11.0791</td>
</tr>
<tr>
<td>73 10.1386</td>
<td>74 6.3158</td>
<td>75 3.6375</td>
<td>76 2.6561</td>
</tr>
<tr>
<td>77 1.8026</td>
<td>78 1.0103</td>
<td>79 1.0693</td>
<td>80 2.3950</td>
</tr>
<tr>
<td>81 4.0822</td>
<td>82 4.6221</td>
<td>83 4.0672</td>
<td>84 3.8460</td>
</tr>
<tr>
<td>85 4.8489</td>
<td>86 6.3964</td>
<td>87 6.4762</td>
<td>88 4.9457</td>
</tr>
<tr>
<td>89 4.4444</td>
<td>90 5.2131</td>
<td>91 5.0389</td>
<td>92 4.6141</td>
</tr>
<tr>
<td>93 5.8722</td>
<td>94 7.9268</td>
<td>95 7.9486</td>
<td>96 5.7854</td>
</tr>
<tr>
<td>97 4.5495</td>
<td>98 5.2696</td>
<td>99 6.3893</td>
<td>100 6.5216</td>
</tr>
<tr>
<td>101 6.2129</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>