

NAG Library Function Document

nag_median_1var (g07dac)

1 Purpose

nag_median_1var (g07dac) finds the median, median absolute deviation, and a robust estimate of the standard deviation for a set of ungrouped data.

2 Specification

```
#include <nag.h>
#include <nagg07.h>

void nag_median_1var (Integer n, const double x[], double y[], double *xme,
                     double *xmd, double *xsd, NagError *fail)
```

3 Description

The data consists of a sample of size n , denoted by x_1, x_2, \dots, x_n , drawn from a random variable X . nag_median_1var (g07dac) first computes the median

$$\theta_{\text{med}} = \text{med}_i \{x_i\}$$

and from this the median absolute deviation can be computed,

$$\sigma_{\text{med}} = \text{med}_i \{|x_i - \theta_{\text{med}}|\}.$$

Finally, a robust estimate of the standard deviation is computed,

$$\sigma'_{\text{med}} = \sigma_{\text{med}} / \Phi^{-1}(0.75)$$

where $\Phi^{-1}(0.75)$ is the value of the inverse standard Normal function at the point 0.75. nag_median_1var (g07dac) is based upon the algorithm used in the function LTMDDV in the ROBETH library, see Marazzi (1987).

4 References

Huber P J (1981) *Robust Statistics* Wiley

Marazzi A (1987) Subroutines for robust and bounded influence regression in ROBETH *Cah. Rech. Doc. IUMSP, No. 3 ROB 2* Institut Universitaire de Médecine Sociale et Préventive, Lausanne

5 Arguments

- | | | |
|----|---|---------------|
| 1: | n – Integer | <i>Input</i> |
| | <i>On entry:</i> the number of observations, n . | |
| | <i>Constraint:</i> n > 1. | |
| 2: | x[n] – const double | <i>Input</i> |
| | <i>On entry:</i> the vector of observations, x_1, x_2, \dots, x_n . | |
| 3: | y[n] – double | <i>Output</i> |
| | <i>On exit:</i> the observations sorted into ascending order. | |

- 4: **xme** – double * *Output*
 On exit: the median, θ_{med} .
- 5: **xmd** – double * *Output*
 On exit: the median absolute deviation, σ_{med} .
- 6: **xsd** – double * *Output*
 On exit: the robust estimate of the standard deviation, σ'_{med} .
- 7: **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_INT_ARG_GT

On entry, **n** = $\langle \text{value} \rangle$.
 Constraint: **n** $\leq \langle \text{value} \rangle$.

NE_INT_ARG_LE

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

7 Accuracy

The computations are believed to be stable.

8 Parallelism and Performance

nag_median_lvar (g07dac) is not threaded in any implementation.

9 Further Comments

nag_median_lvar (g07dac) may be called with the same actual array supplied for arguments **x** and **y**, in which case the sorted data values will overwrite the original contents of **x**.

10 Example

The following program reads in a set of data consisting of eleven observations of a variable **x**. The median, median absolute deviation and a robust estimate of the standard deviation are calculated and printed along with the sorted data in output array **y**.

10.1 Program Text

```
/* nag_median_lvar (g07dac) Example Program.
 *
 * NAGPRODCODE Version.
 *
 * Copyright 2016 Numerical Algorithms Group.
 *
 * Mark 26, 2016.
 *
 */

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

```

#include <nag_stdlib.h>
#include <nagg07.h>

int main(void)
{
    Integer exit_status = 0, i, n;
    NagError fail;
    double *x = 0, xmd, xme, xsd, *y = 0;

    INIT_FAIL(fail);

    printf("nag_median_lvar (g07dac) Example Program Results\n");
    /* Skip heading in data file */
#ifdef _WIN32
    scanf_s("%*[\n]");
#else
    scanf("%*[\n]");
#endif
#ifdef _WIN32
    scanf_s("%" NAG_IFMT " ", &n);
#else
    scanf("%" NAG_IFMT " ", &n);
#endif
    if (n > 1) {
        if (!(x = NAG_ALLOC(n, double)) || !(y = NAG_ALLOC(n, double)))
        {
            printf("Allocation failure\n");
            exit_status = -1;
            goto END;
        }
    }
    else {
        printf("Invalid n.\n");
        exit_status = 1;
        return exit_status;
    }
    for (i = 0; i < n; ++i)
#ifdef _WIN32
        scanf_s("%lf", &x[i]);
#else
        scanf("%lf", &x[i]);
#endif
    /* nag_median_lvar (g07dac).
     * Robust estimation, median, median absolute deviation,
     * robust standard deviation
     */
    nag_median_lvar(n, x, y, &xme, &xmd, &xsd, &fail);
    if (fail.code != NE_NOERROR) {
        printf("Error from nag_median_lvar (g07dac).\n%s\n", fail.message);
        exit_status = 1;
        goto END;
    }

    printf("Output y:\n");
    for (i = 0; i < n; ++i)
        printf("%6.3f%s", y[i], (i % 11 == 10 || i == n - 1) ? "\n" : " ");
    printf("\nxme = %6.3f, xmd = %6.3f, xsd = %6.3f\n", xme, xmd, xsd);
END:
    NAG_FREE(x);
    NAG_FREE(y);
    return exit_status;
}

```

10.2 Program Data

```

nag_median_lvar (g07dac) Example Program Data
11
13.0 11.0 16.0 5.0 3.0 18.0 9.0 8.0 6.0 27.0 7.0

```

10.3 Program Results

nag_median_lvar (g07dac) Example Program Results

Output y:

3.000 5.000 6.000 7.000 8.000 9.000 11.000 13.000 16.000 18.000 27.000

xme = 9.000, xmd = 4.000, xsd = 5.930
