NAG Library Function Document

nag_summary_stats_onevar_combine (g01auc)

1 Purpose

nag_summary_stats_onevar_combine (g01auc) combines sets of summaries produced by nag_summary_stats_onevar (g01atc).

2 Specification

```
#include <nag.h>
#include <nagg01.h>
void nag_summary_stats_onevar_combine (Integer b, const double mrcomm[],
     Integer *pn, double *xmean, double *xsd, double *xskew, double *xkurt,
     double *xmin, double *xmax, double rcomm[], NagError *fail)
```

3 Description

Assume a dataset containing n observations, denoted by $x = \{x_i : i = 1, 2, \ldots, n\}$ and a set of weights, $w = \{w_i : i = 1, 2, \ldots, n\}$, has been split into b blocks, and each block summarised via a call to nag_summary_stats_onevar (g01atc). Then nag_summary_stats_onevar_combine (g01auc) takes the b communication arrays returned by nag_summary_stats_onevar (g01atc) and returns the mean ($\bar{x}$), standard deviation (s_2), coefficients of skewness (s_3) and kurtosis (s_4), and the maximum and minimum values for the whole dataset.

For a definition of $\bar{x}, s_2, s_3$ and s_4 see Section 3 in nag_summary_stats_onevar (g01atc).

4 References

West D H D (1979) Updating mean and variance estimates: An improved method *Comm. ACM* **22** 532–555

5 Arguments

1: **b** – Integer *Input*

On entry: b, the number of blocks the full dataset was split into.

Constraint: **b** ≥ 1.

2: **mrcomm**[**20** $\times$ **b**] – const double *Communication Array*

Note: where **MRCOMM**(i, j) appears in this document, it refers to the array element **mrcomm**$[(j-1) \times 20 + i - 1]$.

On entry: the jth column of **MRCOMM** must contain the information returned in **rcomm** from one of the runs of nag_summary_stats_onevar (g01atc).

3: **pn** – Integer * *Output*

On exit: the number of valid observations, that is the number of observations with $w_i > 0$, for $i = 1, 2, \ldots, n$.

4: **xmean** – double * *Output*

On exit: $\bar{x}$, the mean.

5: **xsd** – double * *Output*

On exit: s_2, the standard deviation.

6: **xskew** – double * *Output*

On exit: s_3, the coefficient of skewness.

7: **xkurt** – double * *Output*

On exit: s_4, the coefficient of kurtosis.

8: **xmin** – double * *Output*

On exit: the smallest value.

9: **xmax** – double * *Output*

On exit: the largest value.

10: **rcomm**[**20**] – double *Communication Array*

On exit: an amalgamation of the information held in **mrcomm**. This is in the same format as **rcomm** from nag_summary_stats_onevar (g01atc).

If **rcomm** is **NULL**, **rcomm** is not referenced.

11: **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_ALLOC_FAIL

Dynamic memory allocation failed.
See Section 2.3.1.2 in How to Use the NAG Library and its Documentation for further information.

NE_BAD_PARAM

On entry, argument ⟨*value*⟩ had an illegal value.

NE_CASES_ONE

On exit we were unable to calculate **xsd**, **xskew** or **xkurt**. A value of 0 has been returned.

NE_CASES_ZERO

On entry, the number of valid observations is zero.

NE_ILLEGAL_COMM

On entry, **mrcomm** is not in the expected format.

NE_INT

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

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 2.7.6 in How to Use the NAG Library and its Documentation for further information.

NE_NO_LICENCE

Your licence key may have expired or may not have been installed correctly.
See Section 2.7.5 in How to Use the NAG Library and its Documentation for further information.

NE_ZERO_VARIANCE

On exit we were unable to calculate **xskew** or **xkurt**. A value of 0 has been returned.

7 Accuracy

Not applicable.

8 Parallelism and Performance

nag_summary_stats_onevar_combine (g01auc) is not threaded in any implementation.

9 Further Comments

The order that the *b* communication arrays are stored in **mrcomm** is arbitrary. Different orders can lead to slightly different results due to numerical accuracy of floating-point calculations.

Both nag_summary_stats_onevar_combine (g01auc) and nag_summary_stats_onevar (g01atc) consolidate results from multiple summaries. Whereas the former can only be used to combine summaries calculated sequentially, the latter combines summaries calculated in an arbitrary order allowing, for example, summaries calculated on different processing units to be combined.

10 Example

This example summarises some simulated data. The data is supplied in three blocks, the first consisting of 21 observations, the second 51 observations and the last 28 observations. Summaries are produced for each block of data separately and then an overall summary is produced.

10.1 Program Text

```
/* nag_summary_stats_onevar_combine (g01auc) Example Program.
 *
 * NAGPRODCODE Version.
 *
 * Copyright 2016 Numerical Algorithms Group.
 *
 * Mark 26, 2016.
 */

#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nagg01.h>

#define MRCOMM(I, J) mrcomm[(J)*20+(I)]

int main(void)
{
  /* Integer scalar and array declarations */
  Integer b, i, iwt, j, nb, pn;
  Integer exit_status = 0;

  /* NAG structures and types */
  NagError fail;

  /* Double scalar and array declarations */
```

```
  double xkurt, xmax, xmean, xmin, xsd, xskew;
  double rcomm[20];
  double *mrcomm = 0, *wt = 0, *x = 0;

  /* Initialize the error structure */
  INIT_FAIL(fail);

  printf("nag_summary_stats_onevar_combine (g01auc) Example Program Results"
         "\n\n");

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

  /* Read in the number of block of data we have */
#ifdef _WIN32
  scanf_s("%" NAG_IFMT "%*[^\n] ", &b);
#else
  scanf("%" NAG_IFMT "%*[^\n] ", &b);
#endif

  if (!(mrcomm = NAG_ALLOC(20 * b, double)))
  {
    printf("Allocation failure\n");
    exit_status = -1;
    goto END;
  }

  /* Loop over each block of data */
  for (j = 0; j < b; j++) {
    /* Read in the number of observations in this block and a flag indicating
       whether weights have been supplied (iwt = 1) or not (iwt = 0) */
#ifdef _WIN32
    scanf_s("%" NAG_IFMT "%" NAG_IFMT "%*[^\n] ", &nb, &iwt);
#else
    scanf("%" NAG_IFMT "%" NAG_IFMT "%*[^\n] ", &nb, &iwt);
#endif

    /* Reallocate X to the required size */
    NAG_FREE(x);
    if (!(x = NAG_ALLOC(nb, double)))
    {
      printf("Allocation failure\n");
      exit_status = -2;
      goto END;
    }

    /* Read in the data for this block */
    if (iwt) {
      /* Weights supplied, so reallocate WT to the required size */
      NAG_FREE(wt);
      if (!(wt = NAG_ALLOC(nb, double)))
      {
        printf("Allocation failure\n");
        exit_status = -3;
        goto END;
      }
      for (i = 0; i < nb; i++)
#ifdef _WIN32
        scanf_s("%lf%lf", &x[i], &wt[i]);
#else
        scanf("%lf%lf", &x[i], &wt[i]);
#endif
    }
    else {
      /* No weights */
      NAG_FREE(wt);
      wt = 0;
```

```
      for (i = 0; i < nb; i++)
#ifdef _WIN32
        scanf_s("%lf", &x[i]);
#else
        scanf("%lf", &x[i]);
#endif
    }
#ifdef _WIN32
    scanf_s("%*[^\n] ");
#else
    scanf("%*[^\n] ");
#endif

    /* Call nag_summary_stats_onevar (g01atc) to summarise
       this block of data */
    pn = 0;
    nag_summary_stats_onevar(nb, x, wt, &pn, &xmean, &xsd, &xskew, &xkurt,
                             &xmin, &xmax, rcomm, &fail);
    if (fail.code != NE_NOERROR && fail.code != NE_CASES_ONE &&
        fail.code != NE_ZERO_VARIANCE && fail.code != NE_CASES_ZERO) {
      printf("Error from nag_summary_stats_onevar (g01atc).\n%s\n",
             fail.message);
      exit_status = 1;
      goto END;
    }

    /* Save RCOMM for future reference */
    for (i = 0; i < 20; i++)
      MRCOMM(i, j) = rcomm[i];

    /* Display the results for this block */
    printf(" Summary for block %" NAG_IFMT "\n", j + 1);
    if (fail.code == NE_CASES_ZERO)
      printf(" No valid observations supplied. All weights are zero.\n");
    else {
      printf(" %" NAG_IFMT " valid observations\n", pn);
      printf("   Mean           %13.2f\n", xmean);
      if (fail.code == NE_CASES_ONE) {
        printf("   Unable to calculate the standard deviation,");
        printf("skewness or kurtosis\n");
      }
      else {
        printf("   Std devn       %13.2f\n", xsd);
        if (fail.code == NE_ZERO_VARIANCE)
          printf("   Unable to calculate the skewness and kurtosis\n");
        else {
          printf("   Skewness       %13.2f\n", xskew);
          printf("   Kurtosis       %13.2f\n", xkurt);
        }
      }
      printf("   Minimum        %13.2f\n", xmin);
      printf("   Maximum        %13.2f\n\n", xmax);
    }
  }

  /* Call nag_summary_stats_onevar_combine (g01auc) to combine the
     summaries across all the blocks */
  nag_summary_stats_onevar_combine(b, mrcomm, &pn, &xmean, &xsd, &xskew,
                                   &xkurt, &xmin, &xmax, rcomm, &fail);
  if (fail.code != NE_NOERROR) {
    printf("Error from nag_summary_stats_onevar_combine (g01auc).\n%s\n",
           fail.message);
    exit_status = 2;
    goto END;
  }

  /* Display the combined results */
  printf(" Summary for the combined data\n");
  if (fail.code == NE_CASES_ZERO)
    printf(" No valid observations supplied. All weights are zero.\n");
```

```
  else {
    printf(" %" NAG_IFMT " valid observations\n", pn);
    printf("   Mean           %13.2f\n", xmean);
    if (fail.code == NE_CASES_ONE) {
      printf("   Unable to calculate the standard deviation,");
      printf(" skewness or kurtosis\n");
    }
    else {
      printf("   Std devn       %13.2f\n", xsd);
      if (fail.code == NE_ZERO_VARIANCE)
        printf("   Unable to calculate the skewness and kurtosis\n");
      else {
        printf("   Skewness       %13.2f\n", xskew);
        printf("   Kurtosis       %13.2f\n", xkurt);
      }
    }
    printf("   Minimum        %13.2f\n", xmin);
    printf("   Maximum        %13.2f\n", xmax);
  }

END:
  NAG_FREE(x);
  NAG_FREE(wt);
  NAG_FREE(mrcomm);

  return (exit_status);
}
```

10.2 Program Data

```
nag_summary_stats_onevar_combine (g01auc) Example Program Data
3                                   :: b
21 1                                :: nb,iwt (1st block)
 -0.62 4.91     -1.92 0.25
 -1.72 3.90     -6.35 3.75
  2.00 1.17      7.65 3.19
  6.15 2.66      3.81 0.02
  4.87 3.59     -0.51 3.63
  6.88 4.83     -5.85 3.72
 -0.72 1.72      0.66 0.78
  2.23 4.74     -1.61 1.72
 -0.15 3.94     -1.15 1.33
 -8.74 0.51     -3.94 2.40
  3.61 3.90                          :: End of x,wt for 1st block
  51 0                               :: nb,iwt (2nd block)
 -0.66 -2.39 -6.25  1.23  2.27 -2.27
 10.12  8.29 -2.99  8.71 -0.74  0.02
  1.22  1.70  4.30  2.99 -0.83 -1.00
  6.57  2.32 -3.47 -1.41 -5.26  0.53
  1.80  4.79 -3.04  1.20 -3.21 -3.75
  0.86  1.27 -5.95 -5.27  1.63  3.59
 -0.01 -1.38 -4.71 -4.82  3.55  0.46
  2.57  1.76 -4.05  1.23 -1.99  3.20
 -0.65  8.42 -6.01                   :: End of x for 2nd block
  28 0                               :: nb,iwt (3rd block)
  1.13 -8.86  5.92 -1.71 -3.99  6.57
 -2.01 -2.29 -1.11  7.14  4.84 -4.44
 -3.32 10.25 -2.11  8.02 -7.31  2.80
 -1.20  1.01  1.37 -2.28  1.28 -3.95
  3.43 -0.61  4.85 -0.11             :: End of x for 3rd block
```

10.3 Program Results

```
nag_summary_stats_onevar_combine (g01auc) Example Program Results

 Summary for block 1
 21 valid observations
   Mean                   0.73
   Std devn               4.40
   Skewness              -0.05
```

```
  Kurtosis               -1.00
  Minimum                -8.74
  Maximum                 7.65

Summary for block 2
51 valid observations
  Mean                    0.28
  Std devn                3.96
  Skewness                0.46
  Kurtosis               -0.16
  Minimum                -6.25
  Maximum                10.12

Summary for block 3
28 valid observations
  Mean                    0.48
  Std devn                4.65
  Skewness                0.19
  Kurtosis               -0.58
  Minimum                -8.86
  Maximum                10.25

Summary for the combined data
100 valid observations
  Mean                    0.51
  Std devn                4.24
  Skewness                0.18
  Kurtosis               -0.59
  Minimum                -8.86
  Maximum                10.25
```