STAT Module Procedures

Andrew C. Hawkins

January 10, 2014

Contents

1 Central Tendency

1.1 ARITHMETIC_MEAN

Description:
      Returns the arithmetic mean of the given data set.
Standard:
      Fortran 95 and later
Class:
      Function
Syntax:
      `RESULT = ARITHMETIC_MEAN(DATASET, LENGTH)`

Arguments:

DATASET	Shall be an array of type `REAL`.
LENGTH	Shall be a scalar of type `INTEGER`.

Return value:

The return value is of type `REAL`.

Example:

```
PROGRAM test_arithmetic_mean
  DOUBLE PRECISION :: x(5) = (/ 1, 2, 3, 4, 5 /)
  PRINT*, ARITHMETIC_MEAN(x, 5)
END PROGRAM
```

1.2 GEOMETRIC_MEAN

Description:

Returns the geometric mean of the given data set.

Standard:

Fortran 95 and later

Class:

Function

Syntax:

RESULT = GEOMETRIC_MEAN(DATASET, LENGTH)

Arguments:

DATASET	Shall be an array of type `REAL`.
LENGTH	Shall be a scalar of type `INTEGER`.

Return value:

The return value is of type `REAL`.

Example:

```
PROGRAM test_geometric_mean
  DOUBLE PRECISION :: x(5) = (/ 1, 2, 3, 4, 5 /)
  PRINT*, GEOMETRIC_MEAN(x, 5)
END PROGRAM
```

1.3 GENERALIZED_MEAN

Description:

Returns the generalized mean of the given data set.

Standard:

Fortran 95 and later

Class:

Function

Syntax:

RESULT = GENERALIZED_MEAN(DATASET, LENGTH, POWER)

Arguments:

DATASET	Shall be an array of type `REAL`.
LENGTH	Shall be a scalar of type `INTEGER`.
POWER	Shall be a scalar of type `INTEGER`.

Return value:

The return value is of type `REAL`.

Example:

```
PROGRAM test_generalized_mean
  DOUBLE PRECISION :: x(5) = (/ 1, 2, 3, 4, 5 /)
  PRINT*, GENERALIZED_MEAN(x, 5, 3)
END PROGRAM
```

1.4 HARMONIC_MEAN

Description:

Returns the harmonic mean of the given data set.

Standard:

Fortran 95 and later

Class:

Function

Syntax:

RESULT = HARMONIC_MEAN(DATASET, LENGTH)

Arguments:

DATASET	Shall be an array of type `REAL`.
LENGTH	Shall be a scalar of type `INTEGER`.

Return value:

The return value is of type `REAL`.

Example:

```
PROGRAM test_harmonic_mean
  DOUBLE PRECISION :: x(5) = (/ 1, 2, 3, 4, 5 /)
  PRINT*, HARMONIC_MEAN(x, 5)
END PROGRAM
```

1.5 MEDIAN

Description:

Returns the median of the given data set.

Standard:

Fortran 95 and later

Class:

Function

Syntax:

RESULT = MEDIAN(DATASET, LENGTH)

Arguments:

DATASET	Shall be an array of type `REAL`.
LENGTH	Shall be a scalar of type `INTEGER`.

Return value:

The return value is of type `REAL`.

Example:

```
PROGRAM test_median
  DOUBLE PRECISION :: x(5) = (/ 1, 2, 3, 4, 5 /)
  PRINT*, MEDIAN(x, 5)
END PROGRAM
```

1.6 MIDRANGE

Description:

Returns the midrange of the given data set.

Standard:

Fortran 95 and later

Class:

Function

Syntax:

RESULT = MIDRANGE(DATASET, LENGTH)

Arguments:

DATASET	Shall be an array of type `REAL`.
LENGTH	Shall be a scalar of type `INTEGER`.

Return value:

The return value is of type `REAL`.

Example:

```
PROGRAM test_midrange
  DOUBLE PRECISION :: x(5) = (/ 1, 2, 3, 4, 5 /)
  PRINT*, MIDRANGE(x, 5)
END PROGRAM
```

1.7   MODE

Description:

Returns the mode of the given data set.

Standard:

Fortran 95 and later

Class:

Function

Syntax:

`RESULT = MODE(DATASET, LENGTH)`

Arguments:

DATASET	Shall be an array of type `REAL`.
LENGTH	Shall be a scalar of type `INTEGER`.

Return value:

The return value is of type `REAL`.

Example:

```
PROGRAM test_mode
  DOUBLE PRECISION :: x(5) = (/ 1, 2, 3, 4, 5 /)
  PRINT*, MODE(x, 5)
END PROGRAM
```

1.8   QUADRATIC_MEAN

Description:

Returns the quadratic mean of the given data set.

Standard:

Fortran 95 and later   Returns the harmonic mean of the given data set.

Class:

Function

Syntax:

`RESULT = QUADRATIC_MEAN(DATASET, LENGTH)`

Arguments:

DATASET	Shall be an array of type `REAL`.
LENGTH	Shall be a scalar of type `INTEGER`.

Return value:

The return value is of type `REAL`.

Example:

```
PROGRAM test_quadratic_mean
  DOUBLE PRECISION :: x(5) = (/ 1, 2, 3, 4, 5 /)
  PRINT*, QUADRATIC_MEAN(x, 5)
END PROGRAM
```

1.9 TRIMEAN

Description:

Returns the trimean of the given data set.

Standard:

Fortran 95 and later

Class:

Function

Syntax:

RESULT = TRIMEAN(DATASET, LENGTH)

Arguments:

DATASET	Shall be an array of type REAL.
LENGTH	Shall be a scalar of type INTEGER.

Return value:

The return value is of type REAL.

Example:

```
PROGRAM test_trimean
  DOUBLE PRECISION :: x(5) = (/ 1, 2, 3, 4, 5 /)
  PRINT*, TRIMEAN(x, 5)
END PROGRAM
```

1.10 WEIGHTED_MEAN

Description:

Returns the weighted mean of the given data set.

Standard:

Fortran 95 and later

Class:

Function

Syntax:

RESULT = WEIGHTED_MEAN(DATASET, WEIGHT, LENGTH)

Arguments:

DATASET	Shall be an array of type REAL.
LENGTH	Shall be a scalar of type INTEGER.

Return value:

The return value is of type REAL.

Example:

```
PROGRAM test_weighted_mean
  DOUBLE PRECISION :: x(5) = (/ 1, 2, 3, 4, 5 /)
  DOUBLE PRECISION :: w(5) = (/ 1, 4, 3, 1, 2 /)
  PRINT*, WEIGHTED_MEAN(x, w, 5)
END PROGRAM
```

2 Statistical Dispersion

2.1 IQRANGE

Description:

Returns the interquartile range of the given data set.

Standard:

Fortran 95 and later

Class:

Function

Syntax:

RESULT = IQRANGE(DATASET, LENGTH)

DATASET	Shall be an array of type `REAL`.
LENGTH	Shall be a scalar of type `INTEGER`.

Return value:

The return value is of type `REAL`.

Example:

```
PROGRAM test_iqrange
  DOUBLE PRECISION :: x(5) = (/ 1, 2, 3, 4, 5 /)
  PRINT*, IQRANGE(x, 5)
END PROGRAM
```

2.2   MEAN_ABSOLUTE_DEVIATION

Description:

Returns the mean absolute deviation of the given data set.

Standard:

Fortran 95 and later

Class:

Function

Syntax:

RESULT = MEAN_ABSOLUTE_DEVIATION(DATASET, LENGTH)

Arguments:

DATASET	Shall be an array of type `REAL`.
LENGTH	Shall be a scalar of type `INTEGER`.

Return value:

The return value is of type `REAL`.

Example:

```
PROGRAM test_mean_absolute_deviation
  DOUBLE PRECISION :: x(5) = (/ 1, 2, 3, 4, 5 /)
  PRINT*, MEAN_ABSOLUTE_DEVIATION(x, 5)
END PROGRAM
```

2.3   MEAN_DIFFERENCE

Description:

Returns the mean difference of the given data set.

Standard:

Fortran 95 and later

Class:

Function

Syntax:

RESULT = MEAN_DIFFERENCE(DATASET, LENGTH)

Arguments:

DATASET	Shall be an array of type `REAL`.
LENGTH	Shall be a scalar of type `INTEGER`.

Return value:

The return value is of type `REAL`.

Example:

```
PROGRAM test_mean_difference
  DOUBLE PRECISION :: x(5) = (/ 1, 2, 3, 4, 5 /)
  PRINT*, MEAN_DIFFERENCE(x, 5)
END PROGRAM
```

2.4   MEDIAN_ABSOLUTE_DEVIATION

Description:

Returns the median absolute deviation of the given data set.

Standard:

Fortran 95 and later

Class:

Function

Syntax:

RESULT = MEDIAN_ABSOLUTE_DEVIATION(DATASET, LENGTH)

Arguments:

DATASET	Shall be an array of type REAL.
LENGTH	Shall be a scalar of type INTEGER.

Return value:

The return value is of type REAL.

Example:

```
PROGRAM test_median_absolute_deviation
  DOUBLE PRECISION :: x(5) = (/ 1, 2, 3, 4, 5 /)
  PRINT*, MEDIAN_ABSOLUTE_DEVIATION(x, 5)
END PROGRAM
```


2.5   RELATIVE_MEAN_DIFFERENCE

Description:

Returns the relative mean difference of the given data set.

Standard:

Fortran 95 and later

Class:

Function

Syntax:

RESULT = RELATIVE_MEAN_DIFFERENCE

Arguments:

DATASET	Shall be an array of type REAL.
LENGTH	Shall be a scalar of type INTEGER.

Return value:

The return value is of type REAL.

Example:

```
PROGRAM test_relative_mean_difference
  DOUBLE PRECISION :: x(5) = (/ 1, 2, 3, 4, 5 /)
  PRINT*, RELATIVE_MEAN_DIFFERENCE(x, 5)
END PROGRAM
```


2.6   SRANGE

Description:

Returns the range of the given data set.

Standard:

Fortran 95 and later

Class:

Function

Syntax:

RESULT = SRANGE(DATASET, LENGTH)

Arguments:

| *DATASET* | Shall be an array of type `REAL`. |
| *LENGTH* | Shall be a scalar of type `INTEGER`. |

Return value:

The return value is of type `REAL`.

Example:

```
PROGRAM test_srange
  DOUBLE PRECISION :: x(5) = (/ 1, 2, 3, 4, 5 /)
  PRINT*, SRANGE(x, 5)
END PROGRAM
```

2.7 STANDARD_DEVIATION

Description:

Returns the standard deviation of the given data set.

Standard:

Fortran 95 and later

Class:

Function

Syntax:

RESULT = STANDARD_DEVIATION(DATASET, LENGTH, CORRECTED)

Arguments:

| *DATASET* | Shall be an array of type `REAL`. |
| *LENGTH* | Shall be a scalar of type `INTEGER`. |

Return value:

The return value is of type `REAL`.

Example:

```
PROGRAM test_standard_deviation
  DOUBLE PRECISION :: x(5) = (/ 1, 2, 3, 4, 5 /)
  PRINT*, STANDARD_DEVIATION(x, 5)
END PROGRAM
```

2.8 VARIANCE

Description:

Returns the variance of the given data set.

Standard:

Fortran 95 and later

Class:

Function

Syntax:

RESULT = VARIANCE(DATASET, LENGTH, CORRECTED)

Arguments:

| *DATASET* | Shall be an array of type `REAL`. |
| *LENGTH* | Shall be a scalar of type `INTEGER`. |

Return value:

The return value is of type `REAL`.

Example:

```
PROGRAM test_variance
  DOUBLE PRECISION :: x(5) = (/ 1, 2, 3, 4, 5 /)
  PRINT*, VARIANCE(x, 5)
END PROGRAM
```

3 Distribution Shape

3.1 KURTOSIS

Description:

Returns the kurtosis of the given data set.

Standard:

Fortran 95 and later

Class:

Function

Syntax:

`RESULT = KURTOSIS(DATASET, LENGTH)`

Arguments:

DATASET	Shall be an array of type `REAL`.
LENGTH	Shall be a scalar of type `INTEGER`.

Return value:

The return value is of type `REAL`.

Example:

```
PROGRAM test_kurtosis
  DOUBLE PRECISION :: x(5) = (/ 1, 2, 3, 4, 5 /)
  PRINT*, KURTOSIS(x, 5)
END PROGRAM
```

3.2 SKEWNESS

Description:

Returns the skewness of the given data set.

Standard:

Fortran 95 and later

Class:

Function

Syntax:

`RESULT = SKEWNESS(DATASET, LENGTH)`

Arguments:

DATASET	Shall be an array of type `REAL`.
LENGTH	Shall be a scalar of type `INTEGER`.

Return value:

The return value is of type `REAL`.

Example:

```
PROGRAM test_skewness
  DOUBLE PRECISION :: x(5) = (/ 1, 2, 3, 4, 5 /)
  PRINT*, SKEWNESS(x, 5)
END PROGRAM
```

4 Other

4.1 COMBINATION

Description:

Returns the combination of `K` numbers out of `N` numbers.

Standard:

Fortran 95 and later

Class:

Function

Syntax:
```
RESULT = COMBINATION(N, K)
```
Arguments:

N	Shall be a scalar of type `INTEGER`.
K	Shall be a scalar of type `INTEGER`.

Return value:
The return value is of type `REAL`.
Example:

```
PROGRAM test_combination
  PRINT*, COMBINATION(10, 5)
END PROGRAM
```

4.2   FACTORIAL

Description:
Returns the factorial of `N`.
Standard:
Fortran 95 and later
Class:
Function
Syntax:
```
RESULT = FACTORIAL(N)
```
Arguments:

N	Shall be a scalar of type `INTEGER`.

Return value:
The return value is of type `REAL`.
Example:

```
PROGRAM test_factorial
  PRINT*, FACTORIAL(10)
END PROGRAM
```

4.3   PERMUTATION

Description:
Returns the permutation of `K` numbers out of `N` numbers.
Standard:
Fortran 95 and later
Class:
Function
Syntax:
```
RESULT = PERMUTATION(N, K)
```
Arguments:

N	Shall be a scalar of type `INTEGER`.
K	Shall be a scalar of type `INTEGER`.

Return value:
The return value is of type `REAL`.
Example:

```
PROGRAM test_permutation
  PRINT*, PERMUTATION(10, 5)
END PROGRAM
```