

# Statistical Functions for Arrays

By David Adams

Technical Note 06-45

## Overview

---

4th Dimensionにはフィールドに対して使用できる7つの統計関数、**Sum, Average, Min, Max, Std deviation, Variance** そして **Sum squares**が用意されています。しかしこれらの関数はフィールドに対してのみ有効で、**PRINT SELECTION**によるレポート印刷の特定のフェーズでの使用に制限されることもあります。他の状況ではこれらの関数を使用することはできません。幸いなことに、これらの基本的な統計関数を4th Dimensionのコードで簡単に書くことができます。このテクニカルノートに付属するサンプルデータベースには、数値型配列に対して使用できる標準的な統計関数が実装されています。単なる置き換えだけでなく、これらのメソッドは以下のような拡張も加えています:

- ・ 加重平均, 範囲, モード, 頻度数が実装されています。
- ・ 分散と標準偏差を、サンプルまたは母集団全体から計算できます。
- ・ 平方和を、2つの一般的な式を使用して計算できます。

このテクニカルノートでは、実装した統計関数について説明し、さらにそれぞれの関数の背景情報を提供します。

## Statistical Functions

---

### Documentation and Compilation

サンプルデータベース内で、それぞれのメソッドごとにコメントが記述されています。さらに、**ArrayStats Read Me**メソッドにそれぞれのメソッドの概要を記載しています。すべての変数や引数は使用時および**Compiler\_ArrayStats**メソッドで宣言しています。コードは、「自動変数定義は行わない」を含め、コンパイル設定のオプションにかかわらずコンパイル可能です。

### Overview of Calling Requirements

サンプルデータベース内のメソッドは、数値型配列を引数に取り、実数を返します。エラーが発生すると、エラーは変数**ArrayStats\_ErrorCode\_I**に格納されます。エラーコードはいつでも**ArrayStatsError\_Get**メソッドをコールして取得することができます。エラーコードの説明は**ArrayStatsError\_GetDefinition**メソッドを使用して取得可能です。一般的にエラーを避けるために、以下のルールを参考にしてください:

- ・ 数値配列のポインタを渡す
- ・ 配列が少なくともひとつの要素を持つことを確認する

## Method Structure

サンプルデータベース中、それぞれのメソッドは渡される引数を注意深くテストします。

例えば以下の**ArrayStats\_GetMax**メソッドを見てみましょう：

```

C_REAL($0;$max_real)
C_POINTER($1;$array_pointer)

$max_real:=0` Initialize result.

`-----
` Test Parameters
`-----

ArrayStatsError_Set (0)

If (Count parameters>=1)
    $array_pointer:=$1

    Case of
        ¥ (Nil($array_pointer))
            ArrayStatsError_Set (2)` A nil pointer was passed to a statistical array function.

        ¥ ((Type($array_pointer->)#Real array )
            & (Type($array_pointer->)#Integer array )
            & (Type($array_pointer->)#LongInt array ))
            ArrayStatsError_Set (3)` A pointer to a non-numeric array was passed.

        ¥ (Size of array($array_pointer->)=0)` This may or may not be considered an error.
            ArrayStatsError_Set (4)` A pointer to an empty array was passed.
    End case
Else
    ArrayStatsError_Set (1)` Not enough parameters were passed to a statistical array function.
End if

If (ArrayStatsError_Get =0)` Array is of acceptable type and has at least 1 element.

    `-----
    ` Run Code of Routine
    `-----

    C_LONGINT($array_size)
    $array_size:=Size of array($array_pointer->)

    $max_real:=$array_pointer->{1}

    C_LONGINT($element)
    For ($element;2;$array_size)
        If ($array_pointer->{$element}>$max_real)
            $max_real:=$array_pointer->{$element}
        End if
    End for
End if

`-----
` Return Result
`-----

$0:=$max_real

```

引数をテストすることによって、不正なポインタが渡された状態で処理を行うことや、

0による割り算を防ぐようになっています。しかし、エラーチェックコードにより、メソッドが長くなっています。コードを修正したり検証したりする際は、以下の点を覚えておいてください:

- ・ 実際はエラーテスト部と機能実装部は分けられます。
- ・ 機能実装部は相対的にシンプルになっています。なぜなら危険な状況をテストする必要がないからです。

このテクニカルノートに記載されたコードは、サンプルデータベースに実装されたエラーテスト部を省略しています。

### **ArrayStats\_GetMean**

(->Non-empty numeric array) : Mean average  
(Pointer) : Real

このメソッドは配列中の値の平均値を返します。コードは以下のとおりです:

```
C_REAL($sum_real)
C_LONGINT($values_count)
$sum_real:=ArrayStats_GetSum ($array_pointer)
$values_count:=Size of array($array_pointer->)
$mean_real:=$sum_real/$values_count
$0:=$mean_real
```

### **ArrayStats\_GetMedian**

(->Non-empty numeric array;{Sort original?}) : Median value  
(Pointer;{Boolean}) : Real

このメソッドは配列中の値の中央値を返します。デフォルトで、このメソッドは元の配列を並び替えます。元の配列を変更したくない場合は、\$2引数に**False**を渡します。しかしこの場合配列が複製されるため、追加のメモリが必要となります。コードは以下のとおりです:

```
$sortOriginal_b:=True

If (Count parameters>=2)
  $sortOriginal_b:=$2
End if

If ($sortOriginal_b)
  SORT ARRAY($array_pointer->) The easiest way to find the midpoint is by sorting the array.
Else ` Leave the original array intact.
  ` The easiest way to figure out the median is with a sorted array.
  ` Since we're not sorting the original, create a temporary copy
  ` for sorting.
  ARRAY REAL(ArrayStats_MedianValuesCopy_ar;0)
  ARRAY INTEGER(ArrayStats_MedianValuesCopy_ai;0)
  ARRAY LONGINT(ArrayStats_MedianValuesCopy_al;0)
```

```

C_LONGINT($array_type)

$array_type:=Type($array_pointer->)

` Copy the values and Update the array pointer for the working code.
Case of
  ¥ ($array_type=Real array )
    COPY ARRAY($array_pointer->;ArrayStats_MedianValuesCopy_ar)
    $array_pointer:=->ArrayStats_MedianValuesCopy_ar

  ¥ ($array_type=Integer array )
    COPY ARRAY($array_pointer->;ArrayStats_MedianValuesCopy_ai)
    $array_pointer:=->ArrayStats_MedianValuesCopy_ai

  ¥ ($array_type=LongInt array )
    COPY ARRAY($array_pointer->;ArrayStats_MedianValuesCopy_al)
    $array_pointer:=->ArrayStats_MedianValuesCopy_al

Else ` Bug in this routine.
  ArrayStatsError_Set (13)` Did not detect non-numeric array type.
End case
End if

If (ArrayStatsError_Get =0)
  C_LONGINT($elements_count)
  $elements_count:=Size of array($array_pointer->)
  C_LONGINT($firstElement_index)
  $firstElement_index:=$elements_count//2

  If (Mod($elements_count;2)=1)` Odd sized.
    C_LONGINT($element)
    $element:=1+$firstElement_index
    $median_real:=$array_pointer->{$element}
  Else
    $median_real:=$array_pointer->{$firstElement_index}
    $median_real:=$median_real+$array_pointer->{$firstElement_index+1}
    $median_real:=$median_real/2
  End if

  If (Not($sortOriginal_b))` The code worked on a temporary copy, clear it out.
    ` Remember: $array_pointer now points to the temporary array.
    DELETE ELEMENT($array_pointer->;1;Size of array($array_pointer->))
  End if
End if

```

## ArrayStats\_GetMin

(->Non-empty numeric array) : Minimum value found  
(Pointer) : Real

このメソッドは配列中の値のうち、最小値を返します。コードは以下のとおりです:

```

C_LONGINT($array_size)

$array_size:=Size of array($array_pointer->)
$min_real:=$array_pointer->{1}

C_LONGINT($element)

For ($element;2;$array_size)
  If ($array_pointer->{$element}<$min_real)
    $min_real:=$array_pointer->{$element}
  End if
End for

$0:=$min_real

```

## ArrayStats\_GetMax

(->Non-empty numeric array) : Maximum value found  
(Pointer) : Real

このメソッドは配列中の値のうち、最大値を返します。コードは以下のとおりです:

```
C_LONGINT($array_size)
$array_size:=Size of array($array_pointer->)
$max_real:=$array_pointer->{1}

C_LONGINT($element)

For ($element;2;$array_size)
  If ($array_pointer->{$element}>$max_real)
    $max_real:=$array_pointer->{$element}
  End if
End for

$0:=$max_real
```

## ArrayStats\_GetMode

(->Non-empty numeric array;->Mode values array;{->Frequency array}) :  
(Pointer;Pointer;{Pointer}) :

このメソッドは配列中の値をスキャンし、最頻値を計算します。結果には0個以上の最頻値が含まれる可能性があるため、このメソッドからの戻り値は\$0ではなく配列に置かれます。追加の引数として戻り値を受け取る2つめの配列を渡すと、それぞれの最頻値ごとに頻度数を計算して返します。例として1つの最頻値を持つケースを見てみましょう:

Input Values	Output Values	Output Counts
1	2	2
2		
5		
2		
4		

最頻値を2つもつケースでは以下ようになります:

Input Values	Output Values	Output Counts
1	2	2
2	5	2
5		
2		
5		

引数として渡す配列タイプと、最頻値を受け取る配列タイプが同じでなければならないことに注意してください。

頻度数を受け取る配列は、数値型配列であればどのタイプでもかまいません。コードは以下のようになります:

```
` Get the unique values from the value arrays along with a count of each time they appear.
ArrayStats_GetFrequencyCounts ($valueArray_pointer;$uniqueValuesArray_pointer;
    ->ArrayStats_ModeUniqueCounts_al)

` Find any modal values.
C_LONGINT($count_min)
C_LONGINT($count_max)

$count_min:=ArrayStats_GetMin (->ArrayStats_ModeUniqueCounts_al)
$count_max:=ArrayStats_GetMax (->ArrayStats_ModeUniqueCounts_al)

If ($count_min=$count_max)` There's no mode.
Else
    C_LONGINT($startFrom_index)
    $startFrom_index:=1

Repeat
    C_LONGINT($modalValue_index)
    $modalValue_index:=Find in array
        (ArrayStats_ModeUniqueCounts_al;$count_max;$startFrom_index)

    If ($modalValue_index>=1)

        $startFrom_index:=$modalValue_index+1` Make next find in array step past current match.
        C_LONGINT($newElementInOutputArrays_index)
        $newElementInOutputArrays_index:=Size of array($modeValuesArray_pointer->)+1
        INSERT ELEMENT($modeValuesArray_pointer->,$newElementInOutputArrays_index;1)
        $modeValuesArray_pointer->{$newElementInOutputArrays_index}:=
            $uniqueValuesArray_pointer->{$modalValue_index}

        If ($returnFrequencyCounts_b)
            $frequencyCountArray_pointer->{$newElementInOutputArrays_index}:=
                ArrayStats_ModeUniqueCounts_al{$modalValue_index}
        End if

    End if
Until ($modalValue_index<=0)
End if
```

### ArrayStats\_GetFrequencyCounts

(->Non-empty numeric array;->Unique values array;->Frequency array) :  
(Pointer;Pointer;Pointer) :

このメソッドは、元の配列に基づき、その配列に含まれる重複しない値の配列と、さらにそれらの頻度数の配列を作成します。基の配列と重複しない値の配列のタイプは同じでなければなりません。最終的な結果は値によって並び替えられます。以下の例題で、このメソッドがどのような結果を返すか見てみましょう:

Input Values	Output Values	Output Counts
1	1	1
2	2	2
5	4	1
2	5	1
4		

コードは以下ようになります:

```
C_LONGINT($sourceArray_size)
C_LONGINT($sourceArray_index)
$sourceArray_size:=Size of array($valueArray_pointer->)

For ($sourceArray_index;1;$sourceArray_size)
  C_LONGINT($positionInUniqueValues_index)
  $positionInUniqueValues_index:=Find in array
    ($uniqueValuesArray_pointer->,$valueArray_pointer->{$sourceArray_index})

  If ($positionInUniqueValues_index<1)
    ` Not listed in the unique values+count arrays yet.
    ` Append items to unique values+count arrays.
    $positionInUniqueValues_index:=Size of array($uniqueValuesArray_pointer->)+1
    INSERT ELEMENT($uniqueValuesArray_pointer->,$positionInUniqueValues_index)
    INSERT ELEMENT($frequencyCountArray_pointer->,$positionInUniqueValues_index)
    $uniqueValuesArray_pointer->{$positionInUniqueValues_index}:=
      ($valueArray_pointer->{$sourceArray_index})
  End if

  $frequencyCountArray_pointer->{$positionInUniqueValues_index}:=
    ($frequencyCountArray_pointer->{$positionInUniqueValues_index})+1
End for

SORT ARRAY($uniqueValuesArray_pointer->,$frequencyCountArray_pointer->)
```

**Note** ArrayStats\_GetFrequencyCounts は ArrayStats\_GetMode で使用されています。参考のためこのテクニカルノートに含めました。

### ArrayStats\_GetRange

(->Non-empty numeric array) : Distance between maximum and minimum  
(Pointer) : Real

このメソッドは、配列に含まれる値のうち、最大値と最小値の差を返します。コードは以下ようになります:

```
C_REAL($min_real)
C_REAL($max_real)
$min_real:=0
$max_real:=0
C_LONGINT($array_size)
$array_size:=Size of array($array_pointer->)
$min_real:=$array_pointer->{1}
$max_real:=$array_pointer->{1}
C_LONGINT($element)

For ($element;2;$array_size)
  If ($array_pointer->{$element}<$min_real)
    $min_real:=$array_pointer->{$element}
  End if
  If ($array_pointer->{$element}>$max_real)
    $max_real:=$array_pointer->{$element}
  End if
End for

$range_real:=$max_real-$min_real

$0:=$range_real
```

## ArrayStats\_GetStandardDeviation

(->Non-empty numeric array;{Type}) : Standard deviation  
(Pointer;{Text}) : Real

このメソッドは、配列に含まれる値の標準偏差を返します。コードは以下のようになります：

```
C_REAL($variance_real)
$variance_real:=ArrayStats_GetVariance ($array_pointer;$type_s)

If (ArrayStatsError_Get =0) ` ArrayStats_GetVariance may set an error not detected above.
  $standardDeviation_real:=$variance_real^0.5` Square root of variance
End if

$0:=$standardDeviation_real
```

オプションの引数を使用すると、セクタを指定することができます：

population

sample

Type引数を省略した場合、デフォルトで"sample"が使用されます。

## Background:

標準偏差は、一連の値の平均分散を求める一般的な方法です。最も簡単で多く使用される標準偏差の計算方法は分散の平方根をとることです。詳細情報については、後述の二乗和の節を参照してください。

## ArrayStats\_GetSum

(->Non-empty numeric array) : Sum  
(Pointer) : Real

このメソッドは、配列中の値の和を返します。コードは以下のようになります：

```
C_LONGINT($values_count)
$values_count:=Size of array($array_pointer->)
C_LONGINT($value_index)

For ($value_index;1;$values_count)
  $sum_real:=$sum_real+$array_pointer->{$value_index}
End for

$0:=$sum_real
```

## ArrayStats\_GetSumSquares

(->Non-empty numeric array) : Sum squares  
(Pointer) : Real



このメソッドは配列中の値の二乗和を返します。コードは以下のようになります:

```
C_REAL($mean_real)
$mean_real:=ArrayStats_GetMean ($array_pointer)
C_LONGINT($array_size)
$array_size:=Size of array($array_pointer->)
C_LONGINT($element)
C_REAL($distanceFromMean_real)

For ($element;1;$array_size)
    $distanceFromMean_real:=$mean_real-$array_pointer->{$element}
    $sumSquares_real:=$sumSquares_real+($distanceFromMean_real*$distanceFromMean_real)
End for

$0:=$sumSquares_real
```

### Background:

二乗和は分散や標準偏差を計算する際に、その計算の基となる、基本的な値です。分散や標準偏差の、分散や標準偏差の背景にあるアイデアは、一連の値中の偏差度数を計ることにあります。数の集合の偏差を計算するには、それぞれの値の差を、平均に対して加算します。残念ながらこの方法では常に0の加算が発生します。(平均から平均の差は0だからです。) この問題を避けるために、平均からそれぞれの値の差は平方がとられます。この方法はより有用な値を出す傾向にあります。分散はおおよそ二乗和の平均であり、標準偏差は分散の二乗和です。

### ArrayStats\_GetSumSquaresAlt

(->Non-empty numeric array) : Sum squares

(Pointer) : Real

このメソッドは配列中の値の二乗和を、別の"計算法"を使用して計算します。このテクニックは手作業で二乗和を計算する際、より簡単に計算できる方法ですが、

**ArrayStats\_GetSumSquares**に実装された方法のように、広く使われていません。この別法は、この方法を使用したシステムで使用する値を計算するために提供されました。コードは以下のようになります:

```
C_REAL($mean_real)
$mean_real:=ArrayStats_GetMean ($array_pointer)

C_REAL($sum_real)
C_REAL($sumSquares_real)
$sum_real:=0
$sumSquares_real:=0

C_LONGINT($array_size)
$array_size:=Size of array($array_pointer->)

C_LONGINT($element)
For ($element;1;$array_size)
    C_REAL($value)
```

```

$value:=$array_pointer->{$element}
$sum_real:=$sum_real+$value
$sumSquares_real:=$sumSquares_real+($value*$value)
End for

$sumSquaresAlt_real:=$sumSquares_real-(($sum_real*$sum_real)/$array_size)

$0:=$sumSquaresAlt_real

```

注 二乗和は分散や標準偏差の計算に使用されます。もし計算方法にこちらのほうを使用するのであれば、`ArrayStats_GetVariance` と `ArrayStats_GetStandardDeviation` を複製するかまたは直接、`ArrayStats_GetSumSquaresAlt` をコールするように変更してください。

## ArrayStats\_GetVariance

(->Non-empty numeric array;{Type}) : Variance  
(Pointer;{Text}) : Real

このメソッドは配列中の値の分散を返します。コードは以下のようになります:

```

C_REAL($sumSquares_real)
$sumSquares_real:=ArrayStats_GetSumSquares ($array_pointer)
C_LONGINT($values_count)
$values_count:=Size of array($array_pointer->)

If ($type_s="sample")
    $variance_real:=$sumSquares_real/($values_count-1)
Else ` population
    $variance_real:=$sumSquares_real/$values_count
End if

$0:=$variance_real

```

オプションの引数を使用する場合、値は以下の2つのうちどちらかとなります:

population

sample

type引数が省略された場合、デフォルトで"sample"が使用されます。

## ArrayStats\_GetWeightedMean

(->Non-empty numeric value array;->Non-empty weighting array) :  
Weighted mean average  
(Pointer;Pointer) : Real

このメソッドは加重平均を返します。コードは以下のようになります:

```

C_LONGINT($elements_count)
$elements_count:=Size of array($valueArray_pointer->)

```

```

C_REAL($weightedValues_sum)
C_REAL($weights_sum)
C_LONGINT($element_index)
$weightedValues_sum:=0
$weights_sum:=0

For ($element_index;1;$elements_count)
  C_REAL($weight)
  $weight:=$weightingArray_pointer->{$element_index}
  $weightedValues_sum:=
    $weightedValues_sum+($valueArray_pointer->{$element_index}*$weight)
  $weights_sum:=$weights_sum+$weight
End for

If ($weights_sum#0)
  $weightedMean_real:=$weightedValues_sum/$weights_sum
Else ` Avoid division by zero. This result may or may not be considered an error.
  ArrayStatsError_Set (8) ` ArrayStats_GetWeightedMean detected a weight sum of zero.
End if

$0:=$weightedMean_real

```

## Background:

加重平均は、一連の値が同程度の重要度を持っていない場合に有用です。例えばいくつかのデータの信頼性が低いと思われる場合や主観的に重要度が低いと思われる場合などです。加重平均を使用する昔からあるアプリケーションは、複数のデータを使用してひとつの等級を計算するものです。例えば、期末レポートは最終評価に対して**40%**を占め、2回行われた通常のテストはそれぞれ**30%**の影響となるといった感じです。

## Error Management Routines

---

### Overview

統計関数は、`nil`ポインタや**0**による割り算など、予想されるエラーを検知し、それをブロックします。エラーを検知するとエラーコードを設定し、関数は**0**を返します。エラー管理関連のメソッドを以下で説明します。

### ArrayStatsError\_Get

(`()`): Error code

(`()`) : Longint

このメソッドはArrayStatsエラーを返します。

### ArrayStatsError\_GetDefinition

(Error code): Error description

(Longint) : Text

このメソッドはArrayStatsエラーコードに対応するエラーメッセージを返します。

以下に定義済みエラーコードとそのメッセージを記述します：

### Code Description

- 1 統計関数に必要な十分な引数が渡されていません。
- 2 統計関数に`nil`ポインタが渡されました。
- 3 統計関数に非数値型の配列が渡されました。
- 4 統計関数に要素数0の配列のポインタが渡されました。
- 5 統計関数にサポートされない比較タイプが渡されました。サポートされるのは 'population' または 'sample' です。
- 6 `sample`が2要素より少ない値の分散を計算することはできません。
- 7 `ArrayStats_GetWeightedMean`にサイズの異なる配列が渡されました。
- 8 `ArrayStats_GetWeightedMean`に渡された加重値の合計が0です。
- 9 `ArrayStatsError_Set`にエラーコードが渡されていません。
- 10 `ArrayStats_GetMode`に渡す基の値配列と最頻値配列は、同じ数値型でなければなりません。
- 11 `ArrayStats_GetMode` 内部バグ：非数値配列が見つかりませんでした。
- 12 `ArrayStats_GetFrequencyCounts`に渡す基の値配列と重複しない値配列は、同じ数値型でなければなりません。
- 13 `ArrayStats_GetMedian` 内部バグ：非数値配列が見つかりませんでした。
- 14 `ArrayStatsError_GetDefinition`にエラーコードが渡されていません。

### ArrayStatsError\_Set

(Error code):

(Longint) :

このメソッドは現在の`ArrayStats`のエラーコードを設定します。

このメソッドは主に`ArrayStats`関数内で使用されます。

### Summary

4th Dimension に含まれる統計関数はフィールドに対する計算にのみ対応していて、しかもしばしば **PRINT SELECTION** 中にものみ利用できます。幸いなことに 4th Dimension のコードを使用して、配列に対して使用できる拡張された統計関数を書くことは難しいことはありません。これらのツールはこのテクニカルノートサンプルデータベースに実装され、またこの文書で説明しました。