平均値と分散の計算

cin と cout および Console の使い方

平均値と分散、標準偏差を求めるプログラムをリスト１に示す。平均値は次式

$$\text{平均値} = \overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i \qquad (1)$$

によって与えられる。分散は

$$\text{分散} = \frac{1}{n}\sum_{i=1}^{n} (X_i - \overline{X})^2$$

で与えられるが、リスト１では次式

$$\sum_{i=1}^{n} (X_i - \overline{X})^2 = \sum_{i=1}^{n} X_i^2 - n \times \overline{X}^2 \qquad (2)$$

を用いて計算している。すなわち、以下のコード

```
double sum = 0.0, ssum = 0.0;
for (int i = 0; i < n; i++){
        sum += x[i];
        ssum += x[i] * x[i];
}
```

によって、和$\text{sum} = \sum X_i$と 2 乗和$\text{ssum} = \sum X_i^2$を求め、式（１）および式（２）を用いて平均値と$X_i - \overline{X}$の 2 乗和を求めている。

標準偏差は分散の平方根

$$\text{分散} = \sqrt{\text{分散}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (X_i - \overline{X})^2}$$

で与えられる。

不偏分散は、分散の推定値として用いられる値であるが、次式

$$\text{不偏分散} = \frac{1}{n-1}\sum_{i=1}^{n} (X_i - \overline{X})^2$$

で算出される値である。

リスト１　平均値と分散を求めるプログラム。

```
#include "stdafx.h"
#include <iostream>

using namespace System;
using namespace std;

int main(array<System::String ^> ^args)
{
        double x[100];
        double c;
        cout << "基準値= ";
        cin >> c;
        int n = -1;
        while (true) {
                n++;
                cout << "data-" << n+1 << " = ";
                cin >> x[n];
                if (x[n] < c) break;
        }
        if (n < 1) return 0;
        double sum = 0.0, ssum = 0.0;
        for (int i = 0; i < n; i++){
                sum += x[i];
                ssum += x[i] * x[i];
        }
        cout << endl << "入力データ..." << endl;
        for (int i = 0; i < n; i++)
                cout << "データ-" << i+1 << " = " << x[i] << endl;

        double mean = sum / n;
        double var = (ssum - n * mean * mean) / n;
        double sd = Math::Sqrt(var);
        double uvar = var * n / (n - 1.0);
        double sd_uvar = Math::Sqrt(uvar);
        cout << endl << "平均値= " << mean << endl;
        cout << "分散= " << var << endl;
        cout << "標準偏差= " << sd << endl;
        cout << "不偏分散= " << uvar << endl;
        cout << "不偏分散の平方根= " << sd_uvar << endl;

        Console::WriteLine();
        Console::WriteLine("Enterキーを押して終了。");
        Console::ReadLine();
        return 0;
}
```

リスト１のプログラムの実行例を図１に示す。プログラムを実行すると「基準値 =」と聞いていくるので、入力データの最小値より小さい値を設定する。基準値を設定して Enter キーを押すと「data-1 =」と聞いてくるの１番目のデータ値を設定する。Enter キーを押す

と「data-2 =」聞いていくるので２番目のデータ値を設定して Enter キーを押す。このデータ値の設定と Enter キーを押す捜査をすべてのデータの入力が終わるまで繰り返す。最後に基準値として設定した値より小さい値を入力したところでデータの入力が終了したと判定される。データの入力後、入力データの表示と計算が行われ、平均値、分散などが出力される。

```
基準値 = -1
data-1 = 1
data-2 = 2
data-3 = 3
data-4 = 4
data-5 = 5
data-6 = 6
data-7 = 7
data-8 = 8
data-9 = 9
data-10 = 10
data-11 = -9

入力データ...
データ-1 = 1
データ-2 = 2
データ-3 = 3
データ-4 = 4
データ-5 = 5
データ-6 = 6
データ-7 = 7
データ-8 = 8
データ-9 = 9
データ-10 = 10

平均値 = 5.5
分散 = 8.25
標準偏差 = 2.87228
不偏分散 = 9.16667
不偏分散の平方根 = 3.02765

Enter キーを押して終了。
```

図１　リスト１のプログラムの実行例。

リスト１は C++標準入出力ストリーム cin と cout を用いたものであるが、Visual C++/CLR の標準入出力用クラス Console を用いて書き直したものをリスト２に示す。

リスト２　Visual C++/CLR 標準入出力用クラス Console を用いたプログラム。

```
#include "stdafx.h"

using namespace System;

int main(array<System::String ^> ^args)
{
        double x[100];
        double c;
        Console::Write("基準値= ");
        c = double::Parse(Console::ReadLine());
        int n = -1;
        while (true) {
                n++;
                Console::Write("data-" + (n+1).ToString() + " = ");
                x[n] = double::Parse(Console::ReadLine());
                if (x[n] < c) break;
        }
        if (n < 1) return 0;
        double sum = 0.0, ssum = 0.0;
        for (int i = 0; i < n; i++){
                sum += x[i];
                ssum += x[i] * x[i];
        }
        Console::WriteLine();
        Console::WriteLine("入力データ...");
        for (int i = 0; i < n; i++)
                Console::WriteLine("データ-" + (i+1).ToString() + " = "
                                        + x[i].ToString());

        double mean = sum / n;
        double var = (ssum - n * mean * mean) / n;
        double sd = Math::Sqrt(var);
        double uvar = var * n / (n - 1.0);
        double sd_uvar = Math::Sqrt(uvar);
        Console::WriteLine();
        Console::WriteLine("平均値= " + mean.ToString());
        Console::WriteLine("分散= " + var.ToString());
        Console::WriteLine("標準偏差= " + sd.ToString());
        Console::WriteLine("不偏分散= " + uvar.ToString());
        Console::WriteLine("不偏分散の平方根= " + sd_uvar.ToString());

        Console::WriteLine();
        Console::WriteLine("Enterキーを押して終了。");
        Console::ReadLine();
        return 0;
}
```

cin および cout を用いた場合と、Console を用いた場合の違いは、以下のとおりである。

入力のときは、cin の場合は、

cin >> c;

というように変数に直接入力する形で書かれる。しかし、Console の場合は、関数 Console;:ReadLine()は、コンソールから 1 行分のデータを文字列として返すものであるので、文字列をその文字列の表す数値に変換してから変数に代入する必要がある。例えば、入力文字列が“123”であれば、その“123”という文字列が 10 進数として表している数値に変換することによって数値として扱うことができる。文字列をその表す数値に変換する関数として Parse がある。“123”という文字列をその表す整数値に変換するときは int の関数 Parse を呼び出して

int::Parse(“123”)

とする。double 型の値に変換するときは、double の関数 Parse を呼び出して、

double::Parse(“1.23”)

などとする。

コンソールから入力した値を変数 c に設定するときは

c = double::Parse(Console::ReadLine());

という形になる。

出力のときは、cout の場合は変数を直接書けばその値がコンソールに表示される。

cout << c;

と書けば、変数 c の値がコンソールに出力されて表示される。

Console クラスを用いるときは、出力の対象はすべて文字列に変換する必要がある。数値データを、その値を表す文字列に変換する関数として ToString がある。変数 c の値を表す文字列に変換するときは

c.ToString()

とすればよい。したがって、変数 c の値をコンソールに出力するときは

Console::WriteLine(“c = “ + c.ToString());

などと書く。