



# RMS

Root Mean Square

Advanced Average

Converting  $\Sigma$  To C++

## THE RMS FORMULA

$$x_{rms} = \sqrt{\frac{1}{n} \sum_{i=0}^{n-1} x_i^2} = \sqrt{\frac{1}{n} (x_0^2 + x_1^2 + \dots + x_{n-1}^2)}$$

## RMS FORMULA TO C++

- $x_i$
- $x_i^2$
- $\sum x_i^2$
- $\sum_{i=0}^{n-1} x_i^2$

```
double rms(double* x, int n)
{
    double sum = 0;

    for (int i = 0; i < n; i++)
        sum += pow(x[i], 2);

    return sqrt(sum / n);
}
```

## NUMBER OF DATA POINTS KNOWN

If the number of data points is known in advance, users enter the number, and the for-loop reads them from the console.

```
int main()
{
    int n = 0;
    cout << "Number of data points?: ";
    cin >> n;
    if (n < 1) exit(1);
    double* x = new double[n];

    for (int i = 0; i < n; i++)
    {
        cout << i << ": ";
        cin >> x[i];
    }

    cout << "rms = " << rms(x, n) << endl;

    delete[] x;

    return 0;
}
```

## NUMBER OF DATA POINTS UNKNOWN

If the number of data points is unknown in advance, the program counts the data as users enter it. Users end input with an end-of-file character.

```
int main()
{
    int n = 0;
    double x[1000];

    while (!cin.eof())
        cin >> x[n++];

    n--;

    if (n < 1) exit(1);

    cout << "rms = " << rms(x, n) << endl;

    return 0;
}
```

## ELIMINATING THE ARRAY

An array makes the rms function possible.

Combining input and calculation eliminates the need for an array.

```
int main()
{
    double    x;
    int       n = 0;
    double    sum = 0;

    for (cin >> x; !cin.eof(); n++)
    {
        sum += pow(x, 2);
        cin >> x;
    }

    if (n < 1) exit(1);

    cout << "rms = " <<
         sqrt(sum / n) << endl;

    return 0;
}
```