Streaming Data Analytics Using Recursive Parameter Estimation

A Study of Recursive Statistics Updates

Nicodème Westphalen

nicodeme.westphalen@forward-college.eu

Forward College

Bachelor of Science in Data Science & Business Analytics

February 12. 2026

Abstract

In streaming environments, data arrive sequentially and key statistics must be updated online. Traditional batch computation of means, variances, and regression coefficients requires repeated recomputation over the full dataset, leading to increasing computational cost. This paper derives numerically stable recursive update formulas for the mean, variance, and the centered cross-deviation terms underlying simple linear regression. By expressing the ordinary least squares estimator entirely in terms of sufficient statistics, we show that regression coefficients can be updated using only quantities available at step n-1 and the newest observation. The resulting formulation reduces complexity, requires only constant memory, and remains algebraically equivalent to batch OLS. These properties make recursive methods particularly suitable for real-time analytics and large-scale streaming applications.

Part I

Recursive Variance and Mean

1 Batch computation: mean and variance

1.1 Mean

The *mean* measures central tendency: the typical value of a dataset. For a dataset $x = \{x_1, \ldots, x_n\}$,

$$\bar{x} = \frac{1}{n}(x_1 + \cdots + x_n) = \frac{1}{n}\sum_{i=1}^{n} x_i. \tag{1}$$

1.2 Variance

The *variance* measures the spread of data points around the mean:

$$s^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2. \tag{2}$$

(Depending on context, the sample variance uses $1/(n-1)$ instead of $1/n$; we return to this distinction later.)

1.3 Batch method

The batch approach is the most intuitive:

1. compute $\bar{x}$ using all datapoints,
2. compute s^2 from $\bar{x}$,
3. if a new datapoint arrives, redo steps (1) and (2) on the enlarged dataset.

1.4 Limitations of batch processing

Batch computation works well when the dataset is fixed and of manageable size. However, it becomes impractical in streaming contexts.

(1) Requires storing all data. All observations must be kept in memory to recompute statistics, which becomes impossible when:

- the dataset is extremely large, or
- data arrive continuously (e.g. sensors, stock prices).

(2) High computational cost. Each new datapoint triggers recomputation of:

- the mean using all data,
- the variance using the mean.

Hence the cost grows with n, making real-time use impractical.

(3) Not suitable for real-time analytics. Batch methods do not update statistics instantaneously when a new datapoint arrives. Streaming scenarios (IoT devices, live dashboards, monitoring systems) require fast, constant-time updates.

2 Conceptualizing recursive updates

When data arrive one-by-one, we do not want to recompute statistics from scratch each time. Recursive formulas update quantities using only:

- the previous statistic,
- the new datapoint,
- a small, constant amount of computation.

This avoids storing past data and enables real-time updates.

3 Recursive mean

Let $\bar{x}_n$ denote the mean after observing n points. Then

$$\bar{x}_n = \frac{x_1 + \cdots + x_n}{n}, \qquad \bar{x}_{n-1} = \frac{x_1 + \cdots + x_{n-1}}{n-1}. \tag{3}$$

Combining these gives the standard streaming update:

$$\bar{x}_n = \frac{(n-1)\bar{x}_{n-1} + x_n}{n} = \bar{x}_{n-1} + \frac{x_n - \bar{x}_{n-1}}{n}. \tag{4}$$

4 A first (naive) approach to the recursive variance

4.1 Setup via sum of squared deviations

A convenient object is the *sum of squared deviations* (SSD later or S_{xx})

$$S_{xx,n} := \sum_{i=1}^{n} (x_i - \bar{x}_n)^2. \tag{5}$$

and from there we have:

$$S_{xx,n-1} := \sum_{i=1}^{n-1} (x_i - \bar{x}_{n-1})^2. \tag{6}$$

Then the population variance is $\sigma_n^2 = S_{xx,n}/n$, and the sample variance is $s_n^2 = S_{xx,n}/(n-1)$.

We split off the last term:

$$S_{xx,n} = \sum_{i=1}^{n-1} (x_i - \bar{x}_n)^2 + (x_n - \bar{x}_n)^2. \tag{7}$$

4.2 Relating deviations around $\bar{x}_n$ and $\bar{x}_{n-1}$

For $i \le n-1$,

$$x_i - \bar{x}_n = (x_i - \bar{x}_{n-1}) + (\bar{x}_{n-1} - \bar{x}_n). \tag{8}$$

Plugging into (6) and expanding yields

$$\begin{aligned} S_{xx,n} &= \sum_{i=1}^{n-1} \Big[(x_i - \bar{x}_{n-1}) + (\bar{x}_{n-1} - \bar{x}_n)\Big]^2 + (x_n - \bar{x}_n)^2 \\ &= \sum_{i=1}^{n-1} (x_i - \bar{x}_{n-1})^2 + 2(\bar{x}_{n-1} - \bar{x}_n) \sum_{i=1}^{n-1} (x_i - \bar{x}_{n-1}) + (n-1)(\bar{x}_{n-1} - \bar{x}_n)^2 + (x_n - \bar{x}_n)^2. \end{aligned} \tag{9}$$

Now use the key identity

$$\sum_{i=1}^{n-1} (x_i - \bar{x}_{n-1}) = 0, \tag{10}$$

so the cross-term disappears and (9) reduces to

$$S_{xx,n} = S_{xx,n-1} + (n-1)(\bar{x}_{n-1} - \bar{x}_n)^2 + (x_n - \bar{x}_n)^2. \tag{11}$$

4.3 Limitations of this approach

The derivation above is a solid first approach: it is correct and clarifies how the new mean affects all previous deviations. It is also a natural point to discuss a practical pitfall: *numerical stability*. The term $(\bar{x}_{n-1} - \bar{x}_n)$ can be very small for large n, and implementations that repeatedly subtract close quantities can suffer from catastrophic cancellation in floating-point arithmetic.

This motivates rewriting (11) into Welford's stable one-pass form, which updates $S_{xx,n}$ using products of deviations rather than fragile differences of means.

5 Deriving the Welford identity

We have like in 4:

$$\bar{x}_n = \bar{x}_{n-1} + \frac{x_n - \bar{x}_{n-1}}{n}. \tag{2}$$

From 4.1 we have the sum of squared deviations (SSD):

$$\begin{aligned} \mathrm{SSD}_n = S_{xx,n} &:= \sum_{i=1}^{n} (x_i - \bar{x}_n)^2 \\ &= \sum_{i=1}^{n-1} (x_i - \bar{x}_n)^2 + (x_n - \bar{x}_n)^2. \end{aligned} \tag{3}$$

And

$$S_{xx,n-1} := \sum_{i=1}^{n-1} (x_i - \bar{x}_{n-1})^2. \tag{4}$$

Now we relate $\bar{x_n}$ and $\bar{x_{n-1}}$:

$$(x_i - \bar{x}_n) = (x_i - \bar{x}_{n-1}) + (\bar{x}_{n-1} - \bar{x}_n). \tag{5}$$

Replace (5) into (3) for the $i = 1, \dots, n-1$ part:

$$\begin{aligned} S_{xx,n} &= \sum_{i=1}^{n-1} \Big[(x_i - \bar{x}_{n-1}) + (\bar{x}_{n-1} - \bar{x}_n)\Big]^2 + (x_n - \bar{x}_n)^2 \\ &= \sum_{i=1}^{n-1} (x_i - \bar{x}_{n-1})^2 + 2(\bar{x}_{n-1} - \bar{x}_n) \sum_{i=1}^{n-1} (x_i - \bar{x}_{n-1}) \\ &\quad + \sum_{i=1}^{n-1} (\bar{x}_{n-1} - \bar{x}_n)^2 + (x_n - \bar{x}_n)^2 \\ &= S_{xx,n-1} + 2(\bar{x}_{n-1} - \bar{x}_n)\Big[\sum_{i=1}^{n-1} x_i - \sum_{i=1}^{n-1} \bar{x}_{n-1}\Big] \\ &\quad + (n-1)(\bar{x}_{n-1} - \bar{x}_n)^2 + (x_n - \bar{x}_n)^2. \end{aligned} \tag{6}$$

But $\sum_{i=1}^{n-1} x_i = (n-1)\bar{x}_{n-1}$, hence the bracket is 0 and the cross-term vanishes:

$$S_{xx,n} = S_{xx,n-1} + (n-1)(\bar{x}_{n-1} - \bar{x}_n)^2 + (x_n - \bar{x}_n)^2. \tag{7}$$

Let

$$\delta_n := x_n - \bar{x}_{n-1} \qquad \Rightarrow \qquad \bar{x}_n = \bar{x}_{n-1} + \frac{\delta_n}{n}. \tag{8}$$

Using (8),

$$\bar{x}_{n-1} - \bar{x}_n = -\frac{\delta_n}{n}, \qquad x_n - \bar{x}_n = x_n - \left(\bar{x}_{n-1} + \frac{\delta_n}{n}\right) = \delta_n\left(1 - \frac{1}{n}\right) = \delta_n \frac{n-1}{n}. \tag{9}$$

Substitute (9) into (7):

$$\begin{aligned} S_{xx,n} &= S_{xx,n-1} + (n-1)\left(\frac{\delta_n}{n}\right)^2 + \left(\delta_n \frac{n-1}{n}\right)^2 \\ &= S_{xx,n-1} + \frac{\delta_n^2}{n^2}\Big[(n-1) + (n-1)^2\Big] \\ &= S_{xx,n-1} + \frac{\delta_n^2}{n^2}(n^2 - n) = S_{xx,n-1} + \delta_n^2 \frac{n-1}{n}. \end{aligned} \tag{10}$$

Finally, from (9),

$$(x_n - \bar{x}_{n-1})(x_n - \bar{x}_n) = \delta_n \cdot \delta_n \frac{n-1}{n} = \delta_n^2 \frac{n-1}{n}. \tag{11}$$

Combining (9) and (10) we get the **Welford identity**:

$$\boxed{S_{xx,n} = S_{xx,n-1} + (x_n - \bar{x}_{n-1})(x_n - \bar{x}_n)}. \tag{12}$$

Then

$$\sigma^2 = \frac{S_{xx,n}}{n} \qquad \text{and} \qquad s^2 = \frac{S_{xx,n}}{n-1} \ (n \geq 2).$$

6 Recursive computation: algorithmic view

6.1 Pseudocode (2.4)

For sake of simplicity we will write $\bar{x} = \mu$

```
# Welford one-pass mean / variance
n = 0
mu = 0
Sxx,n = 0

for x in stream:
    n  = n + 1
    d  = x - mu
    mu = mu + d/n
    d2 = x - mu
    Sxx,n = Sxx,n + d*d2

# population variance: Sxx,n / n
# sample variance:     Sxx,n / (n-1)   if n>1
```

6.2 Flow diagram (2.4)

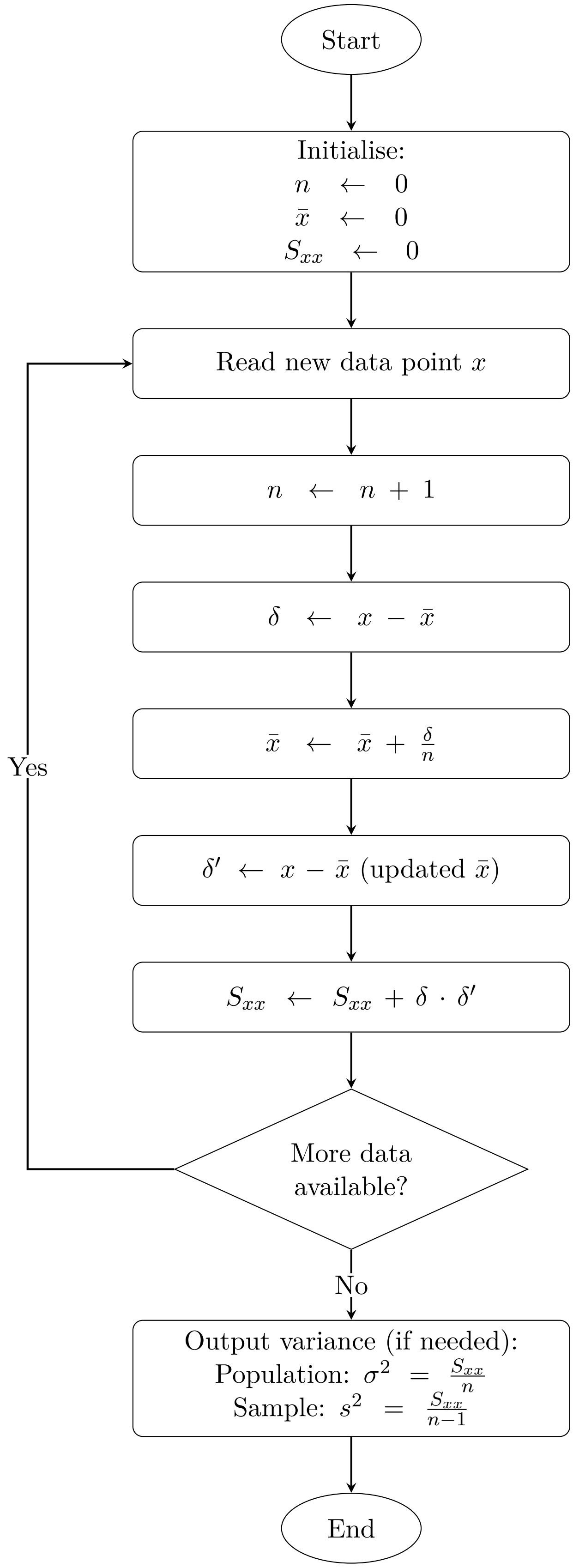

Figure 1: Flow diagram of Welford's recursive updates for mean and variance.

6.3 Manual Simulation (Batch / Non-recursive)

Example Data Stream: 5, 8, 6, 10, 7

Data Point	Batch Mean $\bar{x}_n$	Batch Sample Variance s_n^2
5.000	5.000	–
8.000	6.500	4.500
6.000	6.333	2.333
10.000	7.250	4.917
7.000	7.200	3.700

Table 1: Batch (non-recursive) recomputation of mean and sample variance at each step.

6.4 Manual Simulation

Example Data Stream: 5, 8, 6, 10, 7

Data Point	Updated Mean $\bar{x}_n$	Updated Variance s_n^2
5.000	5.000	-
8.000	6.500	4.500
6.000	6.333	2.333
10.000	7.250	4.917
7.000	7.200	3.700

Table 2: Manual simulation of recursive mean and variance (sample variance $s_n^2 = S_{xx,n}/(n-1)$).

Part II

Recursive Updates for Simple Linear Regression

1 Recap and Motivation

We have

$$\bar{x}_n = \bar{x}_{n-1} + \frac{x_n - \bar{x}_{n-1}}{n} \qquad \text{Mean}$$

and

$$S_{xx,n} = S_{xx,n-1} + (x_n - \bar{x}_{n-1})(x_n - \bar{x}_n) \qquad \text{Sum of squared deviations}$$

$$\Rightarrow \quad s_n^2 = \frac{S_{xx,n}}{n-1} \qquad \text{Sample variance}$$

$$\sigma_n^2 = \frac{S_{xx,n}}{n} \qquad \text{Population variance}$$

Recall:

$$S_{xx,n} := \sum_{i=1}^{n}(x_i - \bar{x}_n)^2, \qquad S_{yy,n} := \sum_{i=1}^{n}(y_i - \bar{y}_n)^2, \qquad S_{xy,n} := \sum_{i=1}^{n}(x_i - \bar{x}_n)(y_i - \bar{y}_n).$$

Note that $S_{xx,n}$ corresponds to the sum of squared deviations it can also be denoted M_n and SSD.

Simple linear regression depends on sample mean, sum of squared deviations S_{xx} and sum of cross-deviation S_{xy}.

If data arrives streaming (which mostly is the case when you use regression to predict), recalculating these costs time and processing power. Recursive updates allow to compute the new statistics solely with the previous statistics and the new datapoint.

2 Simple Least Squares Linear Regression

2.1 Estimators of θ_1 and θ_0

Model:

$$y_i = \theta_0 + \theta_1 x_i + \varepsilon_i$$

we have the residual sum of squares:

$$RSS(\theta_0, \theta_1) = \sum_{i=1}^{n} \varepsilon_i^2 = \sum_{i=1}^{n}(y_i - \theta_0 - \theta_1 x_i)^2$$

We want $RSS(\theta_0, \theta_1)$ to be as small as possible. We choose θ_0 and θ_1 to minimize $RSS(\theta_0, \theta_1)$.

$$\Rightarrow \frac{\partial RSS(\theta 0, n, \theta_{1,n})}{\partial \theta_{0,n}} = -2\sum_{i=1}^{n}\left(y_i - \hat{\theta}_{0,n} - \hat{\theta}_{1,n}x_i\right) = 0 \tag{1}$$

and

$$\frac{\partial RSS(\theta_{0,n}, \theta_{1,n})}{\partial \theta_{1,n}} = -2\sum_{i=1}^{n}\left(x_i y_i - \hat{\theta}_{0,n}x_i - \hat{\theta}_{1,n}x_i^2\right) = 0 \tag{2}$$

From (1) we have:

$$\begin{aligned}
&-2\sum_{i=1}^{n}(y_i - \hat{\theta}_{0,n} - \hat{\theta}_{1,n}x_i) = 0\\
\Rightarrow\ &\sum_{i=1}^{n} y_i - n\hat{\theta}_{0,n} - \hat{\theta}_{1,n}\sum_{i=1}^{n} x_i = 0\\
\Rightarrow\ &\hat{\theta}_{0,n} = \frac{\sum_{i=1}^{n} y_i - \hat{\theta}_{1,n}\sum_{i=1}^{n} x_i}{n}\\
\Rightarrow\ &\hat{\theta}_{0,n} = \frac{n\bar{y} - n\bar{x}_n\hat{\theta}_{1,n}}{n} = \bar{y} - \hat{\theta}_{1,n}\bar{x}_n
\end{aligned} \tag{3}$$

From (2) we have:

$$\begin{aligned}
&-2\sum_{i=1}^{n}(x_i y_i - \hat{\theta}_0 x_i - \hat{\theta}_1 x_i^2) = 0\\
\Rightarrow\ &\sum_{i=1}^{n} x_i y_i - \hat{\theta}_0\sum_{i=1}^{n} x_i - \hat{\theta}_1\sum_{i=1}^{n} x_i^2 = 0
\end{aligned}$$

From (3) we have:

$$\begin{aligned}
&\sum_{i=1}^{n} x_i y_i = (\bar{y}_n - \hat{\theta}_1\bar{x}_n)\sum_{i=1}^{n} x_i + \hat{\theta}_1\sum_{i=1}^{n} x_i^2\\
\Rightarrow\ &\sum_{i=1}^{n} x_i y_i = \bar{y}_n\sum_{i=1}^{n} x_i - \hat{\theta}_{1,n}\bar{x}_n\sum_{i=1}^{n} x_i + \hat{\theta}_{1,n}\sum_{i=1}^{n} x_i^2\\
\Rightarrow\ &\sum_{i=1}^{n} x_i y_i - \bar{y}_n\sum_{i=1}^{n} x_i = \hat{\theta}_{1,n}\left(\sum_{i=1}^{n} x_i^2 - \bar{x}\sum_{i=1}^{n} x_i\right)
\end{aligned}$$

so we can isolate $\hat{\theta}_1$

$$
\begin{aligned}
\Rightarrow\ \hat{\theta}_{1,n} &= \frac{\sum_{i=1}^n x_i y_i - \bar{y}_n \sum_{i=1}^n x_i}{\sum_{i=1}^n x_i^2 - \bar{x}_n \sum_{i=1}^n x_i} \\
&= \frac{\sum_{i=1}^n x_i y_i - n\bar{x}_n \bar{y}}{\sum_{i=1}^n x_i^2 - n\bar{x}_n^2} \\
&= \frac{\sum_{i=1}^n \big(x_i y_i - \bar{y}_n x_i - \bar{x}_n y_i + \bar{x}_n \bar{y}_n\big)}{\sum_{i=1}^n \big(x_i^2 - 2\bar{x}_n x_i + \bar{x}_n^2\big)} \\
\Rightarrow\ \hat{\theta}_{1,n} &= \frac{\sum_{i=1}^n (x_i - \bar{x}_n)(y_i - \bar{y}_n)}{\sum_{i=1}^n (x_i - \bar{x}_n)^2} = \frac{S_{xy,n}}{S_{xx,n}} \qquad (4)
\end{aligned}
$$

Same as for the other statistics, the "batch method" requires to recompute $\sum_{i=1}^n (x_i - \bar{x}_n)(y_i - \bar{y}_n)$ and $\sum_{i=1}^n (x_i - \bar{x}_n)^2$ at every step, thus is computationally heavy.

3 Recursive update for the estimators of θ_0 and θ_1

3.1 Recursive cross-deviation term $S_{xy,n}$

From Part 1, we already have the recursive mean updates:

$$\bar{x}_n = \bar{x}_{n-1} + \frac{x_n - \bar{x}_{n-1}}{n} = \bar{x}_{n-1} + \frac{\delta_{n,x}}{n}, \qquad (1)$$

$$\bar{y}_n = \bar{y}_{n-1} + \frac{y_n - \bar{y}_{n-1}}{n} = \bar{y}_{n-1} + \frac{\delta_{n,y}}{n}, \qquad (2)$$

where we define

$$\delta_{n,x} := x_n - \bar{x}_{n-1}, \qquad \delta_{n,y} := y_n - \bar{y}_{n-1}.$$

Now define the cross-deviation sum

$$S_{xy,n} := \sum_{i=1}^n (x_i - \bar{x}_n)(y_i - \bar{y}_n).$$

$$
\begin{aligned}
S_{xy,n} &= \sum_{i=1}^n (x_i - \bar{x}_n)(y_i - \bar{y}_n) \\
&= \sum_{i=1}^{n-1} (x_i - \bar{x}_n)(y_i - \bar{y}_n) + (x_n - \bar{x}_n)(y_n - \bar{y}_n) \\
&= \sum_{i=1}^{n-1} \Big[(x_i - \bar{x}_{n-1}) + (\bar{x}_{n-1} - \bar{x}_n)\Big]\Big[(y_i - \bar{y}_{n-1}) + (\bar{y}_{n-1} - \bar{y}_n)\Big] + (x_n - \bar{x}_n)(y_n - \bar{y}_n) \\
&= \sum_{i=1}^{n-1} (x_i - \bar{x}_{n-1})(y_i - \bar{y}_{n-1}) + (\bar{y}_{n-1} - \bar{y}_n) \sum_{i=1}^{n-1} (x_i - \bar{x}_{n-1}) \\
&\quad + (\bar{x}_{n-1} - \bar{x}_n) \sum_{i=1}^{n-1} (y_i - \bar{y}_{n-1}) + (n-1)(\bar{x}_{n-1} - \bar{x}_n)(\bar{y}_{n-1} - \bar{y}_n) \\
&\quad + (x_n - \bar{x}_n)(y_n - \bar{y}_n).
\end{aligned}
$$

By definition of the sample means,

$$\sum_{i=1}^{n-1}(x_i - \bar{x}_{n-1}) = 0, \qquad \sum_{i=1}^{n-1}(y_i - \bar{y}_{n-1}) = 0,$$

so the two middle terms vanish. Hence,

$$S_{xy,n} = S_{xy,n-1} + (n-1)(\bar{x}_{n-1} - \bar{x}_n)(\bar{y}_{n-1} - \bar{y}_n) + (x_n - \bar{x}_n)(y_n - \bar{y}_n).$$

Now substitute the mean updates:

$$\bar{x}_{n-1} - \bar{x}_n = -\frac{\delta_{n,x}}{n}, \qquad \bar{y}_{n-1} - \bar{y}_n = -\frac{\delta_{n,y}}{n},$$

and note that

$$y_n - \bar{y}_n = y_n - \left(\bar{y}_{n-1} + \frac{\delta_{n,y}}{n}\right) = \delta_{n,y}\left(1 - \frac{1}{n}\right) = \delta_{n,y}\frac{n-1}{n}.$$

Therefore,

$$\begin{aligned}
S_{xy,n} &= S_{xy,n-1} + (n-1)\left(\frac{\delta_{n,x}}{n}\right)\left(\frac{\delta_{n,y}}{n}\right) + \left(\delta_{n,x}\frac{n-1}{n}\right)\left(\delta_{n,y}\frac{n-1}{n}\right) \\
&= S_{xy,n-1} + \delta_{n,x}\delta_{n,y}\left(\frac{n-1}{n^2} + \frac{(n-1)^2}{n^2}\right) \\
&= S_{xy,n-1} + \delta_{n,x}\delta_{n,y}\left(\frac{(n-1)n}{n^2}\right) \\
&= S_{xy,n-1} + \delta_{n,x}\delta_{n,y}\frac{n-1}{n}.
\end{aligned}$$

Finally, using $y_n - \bar{y}_n = \delta_{n,y}\frac{n-1}{n}$, we obtain the clean form:

$$\boxed{S_{xy,n} = S_{xy,n-1} + (x_n - \bar{x}_{n-1})(y_n - \bar{y}_n)}.$$

$$\begin{aligned}
S_{xy,n} &= S_{xy,n-1} + (\delta_{n,x}\delta_{n,y})\frac{(n-1) + (n^2 - 2n + 1)}{n^2} \\
&= S_{xy,n-1} + (\delta_{n,x}\delta_{n,y})\frac{n^2 - n}{n^2} \\
&= S_{xy,n-1} + (\delta_{n,x}\delta_{n,y})\frac{n(n-1)}{n^2} \\
&= S_{xy,n-1} + (\delta_{n,x}\delta_{n,y})\frac{n-1}{n}.
\end{aligned}$$

And we have (2):

$$x_n - \bar{x}_n = \delta_{n,x}\frac{n-1}{n} \qquad \text{and} \qquad y_n - \bar{y}_n = \delta_{n,y}\frac{n-1}{n}.$$

$$\Rightarrow \quad \delta_{n,x}(y_n - \bar{y}_n) = \delta_{n,x}\delta_{n,y}\frac{n-1}{n}.$$

$$\Rightarrow \quad S_{xy,n} = S_{xy,n-1} + (x_n - \bar{x}_{n-1})(y_n - \bar{y}_n)$$

3.2 Recursive estimators

From 3.1 we have:

$$S_{xy,n} = S_{xy,n-1} + (x_n - \bar{x}_{n-1})(y_n - \bar{y}_n) \tag{1}$$

We also have from I.5 (12), and from recall in 1:

$$S_{xx,n} = S_{xx,n-1} + (x_n - \bar{x}_{n-1})(x_n - \bar{x}_n) \tag{2}$$

From II.2.1 (4) we have

$$\hat{\theta}_1 = \frac{S_{xy,n}}{S_{xx,n}}. \tag{3}$$

Hence

$$(1)(2)(3) \;\Rightarrow\; \quad \hat{\theta}_{1,n} = \frac{S_{xy,n-1} + (x_n - \bar{x}_{n-1})(y_n - \bar{y}_n)}{S_{xx,n-1} + (x_n - \bar{x}_{n-1})(x_n - \bar{x}_n)}.$$

$$\hat{\theta}_{1,n} = \frac{\sum_{i=1}^{n-1}(x_i - \bar{x}_{n-1})(y_i - \bar{y}_{n-1}) + (x_n - \bar{x}_{n-1})(y_n - \bar{y}_n)}{\sum_{i=1}^{n-1}(x_i - \bar{x}_{n-1})^2 + (x_n - \bar{x}_{n-1})(x_n - \bar{x}_n)}.$$

Alternative:

$$\hat{\theta}_{1,n} = \frac{S_{xy,n-1} + \delta_{n,x}\delta_{n,y}\,\frac{n-1}{n}}{S_{xx,n-1} + \delta_{n,x}^2\,\frac{n-1}{n}}.$$

4 Pseudocode

```
# Welford one-pass mean / variance
n = 0
x̄ = 0
ȳ = 0
Sxx,n = 0
Sxy,n = 0

for each new x,y:

    n  = n + 1

    dx = x - x̄
    dy = y - ȳ

    x̄  = x̄  + dx/n
    ȳ  = ȳ  + dy/n

    dx2 = x - x̄
    dy2 = y - ȳ

    Sxx,n = Sxx,n + dx * dx2
    Sxy,n = Sxy,n + dy * dy2

    theta_1e = Sxy,n / Sxx,n
    theta_0e = ȳ - theta_1e * x̄

# Display the estimators
```

5 Flow diagram 2

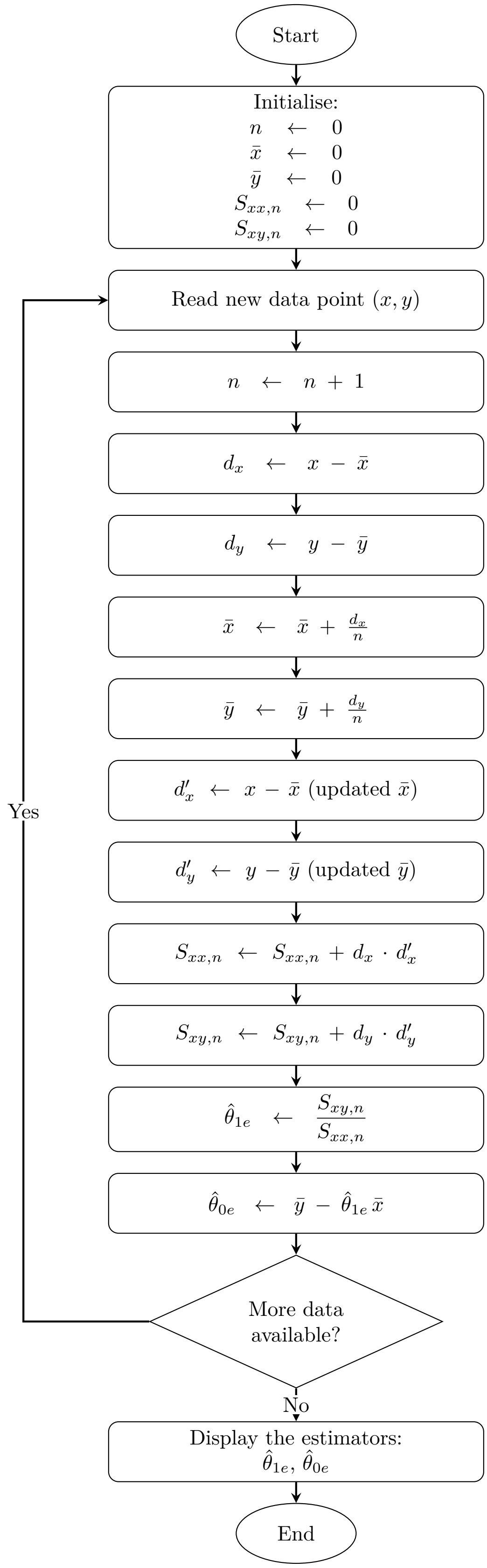

Figure 2: Flow diagram of Welford's recursive updates for estimators $\hat{\theta}_0$ and $\hat{\theta}_1$

6 Manual Simulation (Batch/Non-recursive)

x	y
1	2
2	3
3	5

Table 3: Example dataset for simple linear regression.

Step k	Data point (x_k, y_k)	$\hat{\theta}_{0,k}$	$\hat{\theta}_{1,k}$
1	$(1, 2)$	–	–
2	$(2, 3)$	1	1
3	$(3, 5)$	$\frac{1}{3}$	$\frac{3}{2}$

Table 4: Manual simulation of batch OLS estimates after each additional observation.

n	(x, y)	$\bar{x}$	$\bar{y}$	$S_{xx,n}$	$S_{xy,n}$	$\hat{\theta}_{0e}$	$\hat{\theta}_{1e}$
1	$(1, 2)$	1	2	0	0	–	–
2	$(2, 3)$	1.5	2.5	0.5	0.5	1	1
3	$(3, 5)$	2	$\frac{10}{3}$	2	3	$\frac{1}{3}$	$\frac{3}{2}$

Table 5: Manual simulation of recursive updates for $\bar{x}, \bar{y}, S_{xx,n}, S_{xy,n}, \hat{\theta}_{0,n}and\hat{\theta}_{1,n}$

Part III

Discussion

1 Limitations

The recursive formulation inherits the structural assumptions of classical ordinary least squares. In particular, it assumes that the underlying regression coefficients are constant over time. If the data-generating process shows structural breaks, regime changes, or gradual parameter drift (e.g. flow regimes in fluid dynamics, or inflation in financial modelling), the estimator does not adapt.

Although algebraically equivalent to the batch estimator, the recursive implementation operates in finite-precision arithmetic. Because each update reuses previously stored quantities, small rounding errors propagate through the recursion. Over very long data-streams, these errors may accumulate.

Nonetheless, the recursive formulation remains numerically preferable to naive variance expressions that involve subtracting large, nearly equal quantities.

2 Broader Implications

The derivation highlights that simple linear regression can be reduced to the maintenance of centered sums $S_{xx,n}$ and $S_{xy,n}$. Rather than viewing regression as entailing a repeated global recomputation, it can be interpreted as updating of the statistics.

This conceptual shift connects classical statistical theory with modern streaming and online learning paradigms. It illustrates how traditional estimators can be reformulated to meet the computational constraints of real-time data environments without altering their statistical properties.

Conclusion

This paper derived recursive update formulas for the mean, variance, and cross-deviation terms underlying simple linear regression. By expressing the OLS estimator entirely in terms of centered sum, we showed that regression coefficients can be updated using only previously stored statistics and the newest observation.

The resulting formulation achieves constant memory usage and constant computational complexity while remaining algebraically identical to the batch estimator. These properties make recursive regression particularly suitable for streaming data applications and large-scale computational settings.

Future work could for example be to include a forgetting factor to make up for the piling up errors in the recursive regression, or could extend the framework to other regressions, or robust variants.

References

[1] Welford, B. P. (1962). Note on a method for calculating corrected sums of squares and products. *Technometrics*, 4(3), 419–420.

[2] Haykin, S. (2002). *Adaptive Filter Theory.* Prentice Hall.

[3] Zhang, J., & Wang, D. (2005). Recursive algorithms for real-time statistical estimation. *IEEE Transactions on Signal Processing*, 53(12), 4683–4693.

[4] Hamilton, J. D. (1994). *Time Series Analysis.* Princeton University Press.

[5] Shumway, R. H., & Stoffer, D. S. (2017). *Time Series Analysis and Its Applications: With R Examples.* Springer.

[6] Widrow, B., & Hoff, M. E. (1960). Adaptive switching circuits. *IRE WESCON Convention Record*, 4, 96–104.

[7] Bottou, L. (1998). Online learning and stochastic approximations. In *On-Line Learning in Neural Networks* (pp. 9–42). Cambridge University Press.

[8] Lippmann, R. P. (1987). An introduction to computing with neural nets. *IEEE ASSP Magazine*, 4(2), 4–22.

[9] Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2), 257–286.

[10] Tsay, R. S. (2010). *Analysis of Financial Time Series.* John Wiley & Sons.

[11] Gama, J., & Rodrigues, P. P. (2010). Issues in evaluation of stream learning algorithms. In *Proceedings of the 1st Workshop on Applications of Pattern Analysis.*

[12] Bifet, A., & Gavaldà, R. (2009). Adaptive learning from evolving data streams. In *International Symposium on Intelligent Data Analysis* (pp. 249–260). Springer.

[13] Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018). *Foundations of Machine Learning.* MIT Press.