

Proof that the single-variable linear-regression predictor derived using the general matrix-based multiple regression algorithm gives the same results as the original Pyret implementation.

Given: a set of inputs  $\{x, \dots\}$  and their corresponding outputs  $\{y, \dots\}$ .

Let

$$X = \begin{bmatrix} 1 & x \\ \vdots & \vdots \\ \vdots & \vdots \end{bmatrix}, Y = \begin{bmatrix} y \\ \vdots \\ \vdots \end{bmatrix}$$

Using the multiple-regression algorithm, we get

$$B = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = (X^T X)^{-1} X^T Y. \quad (1)$$

and the predictor function is  $y = \alpha + \beta x$ .

We have

$$X^T = \begin{bmatrix} 1 & \cdot & \cdot & \cdot \\ x & \cdot & \cdot & \cdot \end{bmatrix}$$

$$\therefore X^T X = \begin{bmatrix} 1 & \cdot & \cdot & \cdot \\ x & \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} 1 & x \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \end{bmatrix} = \begin{bmatrix} n & \Sigma x \\ \Sigma x & \Sigma x^2 \end{bmatrix}$$

We then have

$$\det X^T X = n\Sigma x^2 - (\Sigma x)^2 = \Delta \text{ (say)}$$

$$\text{and } \text{cof } X^T X = \begin{bmatrix} \Sigma x^2 & -\Sigma x \\ -\Sigma x & n \end{bmatrix}$$

The adjoint of a matrix is the transpose of its cofactor matrix. So

$$\text{adj } X^T X = (\text{cof } X^T X)^T$$

But  $\text{cof } X^T X$  is diagonally symmetric, so its transpose is itself. So

$$\text{adj } X^T X = \text{cof } X^T X$$

The inverse of a matrix is its adjoint divided by its determinant. So

$$(X^T X)^{-1} = \frac{\text{adj } X^T X}{\Delta} = \left( \frac{1}{\Delta} \right) \begin{bmatrix} \Sigma x^2 & -\Sigma x \\ -\Sigma x & n \end{bmatrix}$$

Putting all this in (1), we have

$$B = \left( \frac{1}{\Delta} \right) \begin{bmatrix} \Sigma x^2 & -\Sigma x \\ -\Sigma x & n \end{bmatrix} \begin{bmatrix} 1 & \cdot & \cdot & \cdot \\ x & \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} y \\ \cdot \\ \cdot \\ \cdot \end{bmatrix}$$

$$= \left( \frac{1}{\Delta} \right) \begin{bmatrix} \Sigma x^2 & -\Sigma x \\ -\Sigma x & n \end{bmatrix} \begin{bmatrix} \Sigma y \\ \Sigma xy \end{bmatrix}$$

$$= \left( \frac{1}{\Delta} \right) \begin{bmatrix} \Sigma x^2 \Sigma y - \Sigma x \Sigma xy \\ -\Sigma x \Sigma y + n \Sigma xy \end{bmatrix}$$

$$\begin{aligned}\therefore \alpha &= \frac{\Sigma x^2 \Sigma y - \Sigma x \Sigma xy}{n \Sigma x^2 - (\Sigma x)^2} \\ \text{and } \beta &= \frac{n \Sigma xy - \Sigma x \Sigma y}{n \Sigma x^2 - (\Sigma x)^2} \quad (2)\end{aligned}$$

Back to the original Pyret implementation. There we have

$$\begin{aligned}\beta &= \frac{\Sigma xy - \frac{\Sigma x \Sigma y}{n}}{\Sigma x^2 - \frac{(\Sigma x)^2}{n}} \\ &= \frac{n \Sigma xy - \Sigma x \Sigma y}{n \Sigma x^2 - (\Sigma x)^2}\end{aligned}$$

$$\begin{aligned}\text{and } \alpha &= \bar{y} - \beta \bar{x} \\ &= \left( \frac{\Sigma y}{n} \right) - \left( \frac{n \Sigma xy - \Sigma x \Sigma y}{n \Sigma x^2 - (\Sigma x)^2} \right) \left( \frac{\Sigma x}{n} \right) \\ &= \left( \frac{\Sigma y}{n} \right) - \left( \frac{n \Sigma x \Sigma xy - (\Sigma x)^2 \Sigma y}{n(n \Sigma x^2 - (\Sigma x)^2)} \right) \\ &= \frac{\Sigma y(n \Sigma x^2 - (\Sigma x)^2) - n \Sigma x \Sigma xy + (\Sigma x)^2 \Sigma y}{n(n \Sigma x^2 - (\Sigma x)^2)} \\ &= \frac{n \Sigma x^2 \Sigma y - (\Sigma x)^2 \Sigma y - n \Sigma x \Sigma xy + (\Sigma x)^2 \Sigma y}{n(n \Sigma x^2 - (\Sigma x)^2)} \\ &= \frac{n \Sigma x^2 \Sigma y - n \Sigma x \Sigma xy}{n(n \Sigma x^2 - (\Sigma x)^2)} \\ &= \frac{\Sigma x^2 \Sigma y - \Sigma x \Sigma xy}{n \Sigma x^2 - (\Sigma x)^2}\end{aligned}$$

But these match exactly the values for  $\alpha, \beta$  in (2). QED.