9/18/18

$$\boxed{V[\Delta y] = E[(\Delta y)^2]}$$

<u>Given:</u> $E[\Delta x_i] = 0$

$\sigma[\Delta x_i] = \sigma_i$

<u>We know:</u> $\sigma[a] = \sqrt{E[a - E[a]]^2} \rightarrow V[a] \rightarrow$ <u>Variance:</u>

~~EXE~~

<u>From linearization we know:</u>

$$\Delta y: \frac{\partial f}{\partial x_1} \Delta x_1 + \dots + \frac{\partial f}{\partial x_n} \Delta x_n$$

$$= c_1 \Delta x_1 + \dots + c_n \Delta x_n, \quad \text{where } c_i = \frac{\partial f}{\partial x_i}$$

$$V[\Delta y] = E[(c_1 \Delta x_1 + \dots + c_n \Delta x_n)^2]$$

$$= E[c_1^2 \Delta x_1 + \dots + c_n^2 (\Delta x_n)^2 + 2 c_1 c_2 \Delta x_1 \Delta_j + \dots$$

<u>We know:</u> $\underline{E[K \cdot a] = K E[a]}$

~~and since~~

$$c_1^2 E[(\Delta x_1)^2] + \dots + c_n^2 E[(\Delta x_n)^2]$$

$$V[\Delta y] = c_1^2 \sigma^2[(\Delta x_1)] + \dots + c_n^2 \sigma^2[(\Delta x_n)].$$

Summarize:

We have: $f(x_1, \dots, x_n)$ – we know the fcn.

$\tilde{x}_1, \dots, \tilde{x}_n$ = approx values from measurements.

$\Delta x_i = \tilde{x}_i - x_i$: $E[\Delta x_i] = 0$, $\boxed{\text{We know } \sigma[\Delta x_i]}$

$\tilde{y} = f(\tilde{x}_1, \dots, \tilde{x}_n)$ – we compute

?

$\Delta y = \tilde{y} - y$, $y \overset{def}{=} f(x_1, \dots, x_n)$

actual values of x_i

Under assumption of linearization.

$E[\Delta y] = 0$

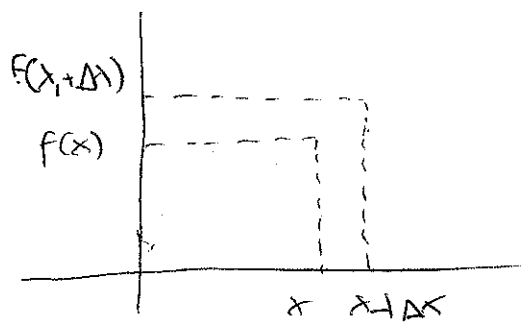
$$\sigma[\Delta y] = \sqrt{\sum_{i=1}^{n} \left(\frac{\partial f}{\partial x_i}\right)^2 \sigma_i^2}$$

Algorithmically:

Case 1: we know $f(x_1, \dots, x_n)$ analytical expression.

⚠ Problem: Often we don't have this analytical expression.

so it's not easy to compute the derivative.



$$\frac{\partial f}{\partial x} = \lim_{x \to 0} \frac{f(x+\Delta x) - f(x)}{\Delta x}$$

$$\frac{\partial f}{\partial x} \approx \frac{f(x+\Delta x) - f(x)}{\Delta x}$$

<u>numerical differentiation</u>

$$\frac{\partial f}{\partial x_i} \frac{f(x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \overset{h_i}{\ldots} x_n) - \tilde{y}}{h_i}$$

<u>Given:</u> $f(x_1, \ldots, x_n)$

$(\tilde{x}_1, \ldots, \tilde{x}_n)$, $\sigma_1, \ldots, \sigma_n$

<u>We want to compute σ</u>

$$\tilde{y} = f(\tilde{x}_1, \ldots, \tilde{x}_n)$$

$$\frac{\partial f}{\partial x_i} = \frac{f(\tilde{x}_1, \ldots, \tilde{x}_{i-1}, \tilde{x}_i + h_i, \tilde{x}_{i+1}, \ldots, \tilde{x}_n) - \tilde{y}}{h_i}$$

$$\sigma = \sqrt{\sum \left(\frac{\partial f}{\partial x_i}\right)^2 \sigma_i^2}$$

$$\sigma = \sqrt{\frac{f(\tilde{x}_1 + h_1, \tilde{x}_2, \ldots, \tilde{x}_n) - \tilde{y})^2}{h_1^2} \sigma_1^2 + \frac{f(\tilde{x}_1, \tilde{x}_2 + h, \ldots, \tilde{x}_n) - \tilde{y})^2}{h_2^2} \sigma_2^2 + \ldots}$$

<u>Trick</u> $h_i = \sigma_i$

$$\sigma = \sqrt{(f(\tilde{x}_1+\sigma_1, \tilde{x}_2, \dots) - \tilde{y})^2 + (f(\tilde{x}_1, \tilde{x}_2+\sigma_2, \dots \tilde{x}_n) - \tilde{y})^2 + \dots}$$

<u>HW:</u>

1) Write a code that computes σ

2) <u>Check:</u> your code by comparing with simulation results.

e.g.

given a fcn.

$f(x_1, \dots, x_n)$ – given a fcn.

$\tilde{x}_1, \dots, \tilde{x}_n$ – values

$\sigma_1, \dots, \sigma_n$ – signms. values

sim. $\Delta x_i = \sigma_i$ gauss()

sim. $x_i = \tilde{x}_i - \Delta x_i$

sim $y = f(x_1, \dots, x_n)$

sim $\Delta y = \tilde{y} - y$

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (\Delta y^{(i)})^2}$$