$y = f(x, \vec{\beta}) = f(x, \beta_1, \beta_2, \beta_3, \dots \beta_n)$

<u>IFF</u> : $\vec{\beta}$ uncorrelated, $\Rightarrow$ $\sigma_y^2 = \sum_i \left(\frac{\partial f(\vec{\beta})}{\partial \beta_i}\right)^2 \sigma_{\beta_i}^2$

<u>But</u> : $\vec{\beta}$ are almost always correlated, i.e., $\sigma_{\beta_i \beta_j} \neq 0$

In that case, we generalize: $\sigma_y^2 = J \cdot \text{cov} \cdot J^T$

where J: the Jacobian $= \left[\frac{\partial y}{\partial \beta_1} \; \frac{\partial y}{\partial \beta_2} \; \frac{\partial y}{\partial \beta_3} \; \cdots\cdots \; \frac{\partial y}{\partial \beta_n} \right]$

Cov: the Covariance matrix $= \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \sigma_{13} & \cdots & \sigma_{1N} \\ \sigma_{21} & \sigma_2^2 & \sigma_{23} & \cdots & \vdots \\ \vdots & & \sigma_3^2 & & \vdots \\ \vdots & & & \ddots & \vdots \\ \sigma_{n1} & \cdots & \cdots & \cdots & \sigma_n^2 \end{bmatrix}$ where $\sigma_{ij} = \sigma_i \cdot \sigma_j$

(this comes from Python's pcov, from curve_fit()! :))

For the Jacobian, no need to calculate $\frac{\partial y}{\partial \beta_i}$ analytically!

$\Rightarrow$ Use numerical approximation:

$$\frac{\partial y}{\partial \beta_i} = \frac{\partial f(x, \vec{\beta})}{\partial \beta_i}$$

$$\approx \frac{f\left(x, \beta_i + \frac{\Delta \beta_i}{2}\right) - f\left(x, \beta_i - \frac{\Delta \beta_i}{2}\right)}{\Delta \beta_i}$$

where $\Delta \beta_i = 10^{-8} \cdot \beta_i$