# vcov method for ascr\_tmb

## input:

* object – the output from a fit().
* types – character, accept any combination from three types: {“linked”, “fitted”, “derived”}, the default is “linked”. If only one type is assigned, the output will be a matrix. If more than one types are assigned, for example “types = c(‘linked’, ‘fitted’)”, the output will be a list with two components, named as ‘linked’ and ‘fitted’ respectively. Also accept ‘all’, which means all of these three types.
* pars – character, accept any conbination from parameters used in the ‘object’. Defaultly is NULL, means all of the parameters involved in the fit function.
* new\_covariates – a data frame with one row and all columns needed for the extended parameters assigned in argument ‘pars’.

## output:

A matrix if only one ‘types’ assigned or a named list if multiple ‘types’ assigned. Output for different types:

* ‘linked’ – Origin covariance matrix based on the output from TMB.
* ‘fitted’ – transformed from ‘linked’ covariance matrix with the reverse link functions by delta method. Details for this component as follows:

*Here is an example, suppose we are fitting an ‘ihd’ model with detect function of ‘hn’ and covariates for ‘D’ are ‘x1’ and ‘y1’. The ‘linked’ covariance matrix (denoted as SIGMA) is a 5 \* 5 matrix for* ***theta*** *= (‘D.(Intercept)’, ‘D.x1’, ‘D.y1’, ‘g0’, ‘sigma’)^T. Another key part is the transform function* ***G****(****theta****) = (g\_1(****theta****), g\_2(****theta****), ..., g\_n(****theta****)).*

*If the ‘new\_covariates’ is not assigned, then G(theta) = (g\_1(D.(Intercept)), g\_2(D.x1), g\_3(D.y1), g\_4(g0), g\_5(sigma)), where g\_1(x) to g\_3(x) are identical functions, and g\_4(x) = exp(x)/(exp(x) + 1), g\_5(x) = exp(x). So the J = d(****G****)/d(****theta****) is a 5 \* 5 diagnal matrix with J\_(1,1) = J\_(2,2) = J\_(3,3) = 1, and J\_(4,4) = exp(g0)/(exp(g0) + 1) ^ 2, and J\_(5,5) = exp(sigma), where the estimations from the model will be used for numerical evaluation.*

*If the ‘new\_covariates’ is assigned, then it should be a data frame contains two new values for ‘D.x1’ and ‘D.y1’, let’s denote them to be ‘v1’ and ‘v2’ respectively. In this case, the* ***G****(****theta****) = (g\_1(D.(Intercept), D.x1, D.y1), g\_2(g0), g\_3(sigma)), where g\_1(...) = exp(D.(Intercept) + D.x1 \* v1 + D.y1 \* v2), and g\_2(x) = exp(x)/(exp(x) + 1), g\_3(x) = exp(x). So the J = d(****G****)/d(****theta****) is a 3 \* 5 matrix, where J\_(i, j) = d(g\_i)/d(theta\_j). With assgined v1 and v2, and estimation of all parameters, the numerical evaluation for J is avaliable.*

*Finally, the ‘fitted’ covariance matrix eqauls to (J %\*% SIGMA %\*% J^T)*

* ‘derived’ – Covariance matrix for ‘esa’