```matlab
function K = covNoise(hyp, x, z, i)

% Independent covariance function, ie "white noise", with specified variance.
%
% Based on the covNoise.m function of the GPML Toolbox -
%   with the following changes:
%       - if the input values x(:,1:end-1) == z(:,1:end-1) and the labels x(:,end) == z(:,end)
%           are equal - label specific noise term will be added
%           (hyp(x(:,2))
%
% The covariance function is specified as:
%
% k(x^p,x^q) = s2 * \delta(p,q)
%
% where s2 is the noise variance and \delta(p,q) is a Kronecker delta function
% which is 1 iff p=q and zero otherwise. Two data points p and q are considered
% equal if their norm is less than 1e-9. The hyperparameter is
%
% hyp = [ log(sqrt(s2)) ]
%
% by Robert Duerichen
% 04/02/2014


tol = 1e-9;  % threshold on the norm when two vectors are considered to be equal
if nargin<2, K = 'nL'; return; end                  % report number of parameters
if nargin<3, z = []; end                                   % make sure, z exists
xeqz = numel(z)==0; dg = strcmp(z,'diag') && numel(z)>0;        % determine mode
if ndims(x)==ndims(z) && all(size(x)==size(z)), xeqz = norm(x-z,'inf')<tol; end

n = size(x,1);
s2 = exp(2.*hyp);                                                % noise variance

% precompute raw
if dg                                                               % vector kxx
  K = ones(n,1);
else
  if xeqz                                                 % symmetric matrix Kxx
    K = eye(n);
  else                                                   % cross covariances Kxz
    K = double(sq_dist(x(:,1:end-1)',z(:,1:end-1)')<tol*tol);
  end
end

if nargin<4                                                        % covariances
  K = diag(s2(x(:,2)))*K;
else                                                               % derivatives
  if i <= length(s2)
    K(diag(x(:,2)==i)) = 2.*s2(i);
    K(diag(x(:,2)~=i)) = 0;

  else
    error('Unknown hyperparameter')
  end

end
```