```
function X = NewtonsMethodTemplate(f, J, x0, tol, N)
    % Returns the iterations of the Newton's method
    % f: Function handle
    %    Objective function, i.e. equation f(x)=0
    % J: Function handle
    %    Jacobian of f
    % x0: Initial root estimate, Nx x 1
    % tol: tolerance
    % N: Maximum number of iterations
    if nargin < 5
        N = 100;
    end
    if nargin < 4
        tol = 1e-10;
    end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Define variables
    % Allocate space for iterations (X)

    Nx = size(x0, 1);
    X = zeros(Nx, N);
    X(:) = nan;

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    xn = x0; % initial estimate
    n = 1; % iteration number
    fn = f(xn); % save calculation
    X(:, n) = xn;
    % Iterate until f(x) is small enough or
    % the maximum number of iterations has been reached
    iterate = norm(fn,Inf) > tol;
    while iterate
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        % Calculate and save next iteration value x

        xn = xn - J(xn) \ fn;
        n = n + 1;
        X(:, n) = xn;

        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        fn = f(xn); % save calculation for next iteration
        % Continue iterating?
        iterate = norm(fn,Inf) > tol && n <= N;
    end

    if norm(fn,Inf) > tol && n > N
        fprintf('Terminated early!\n')
    end

    X(:,~any(~isnan(X), 1))=[];

end
```

```
Not enough input arguments.

Error in NewtonsMethodTemplate (line 20)
    Nx = size(x0, 1);
```

Published with MATLAB® R2019a