```
% Create table

% Vectorize delta t and M
delt = [5/250, 2.5/250, 1/250];
M = [1000, 2000, 4000, 8000, 16000, 32000, 64000];

for j = 1:7
    [Sdcall(j,3),Sdput(j,3)] = Helper_table(M(j),delt(3));
end

% Consider the smallest delta t = 1/250
% Create the initial vectors
call_lower = [];
call_upper = [];
put_lower = [];
put_upper = [];

% Mean value
mean_call = Vcall(:,3);
mean_put = Vput(:,3);

% Then use loop to calculate CI
% CI = mean + c*std/sqrt(N) -- Lower bound
% CI = mean - c*std/sqrt(N) -- Upper bound
for k = 1:7
   lower_cvalue = Vcall(k,3)-1.96*Sdcall(k,3)/sqrt(M(k));
   call_lower = [call_lower,lower_cvalue];
   upper_cvalue = Vcall(k,3)+1.96*Sdcall(k,3)/sqrt(M(k));
   call_upper = [call_upper,upper_cvalue];
   lower_pvalue = Vput(k,3)-1.96*Sdput(k,3)/sqrt(M(k));
   put_lower = [put_lower,lower_pvalue];
   upper_pvalue = Vput(k,3)+1.96*Sdput(k,3)/sqrt(M(k));
   put_upper = [put_upper,upper_pvalue];
end

% Create the table
table_call = table(M',mean_call,call_lower',call_upper');
table_call.Properties.VariableNames = {'M','Mean','Call lower bound','Call upper↙
bound'};
disp(table_call);

table_put = table(M',mean_put,put_lower',put_upper');
table_put.Properties.VariableNames = {'M','Mean','Put lower bound','Put upper bound'};
disp(table_put);
```