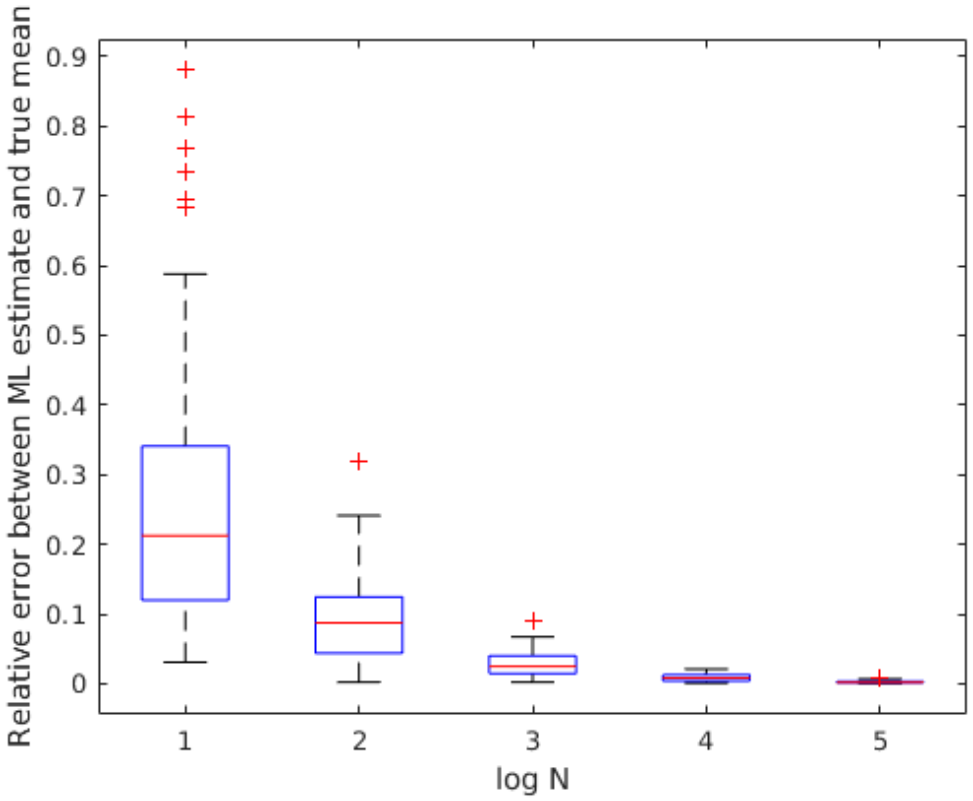
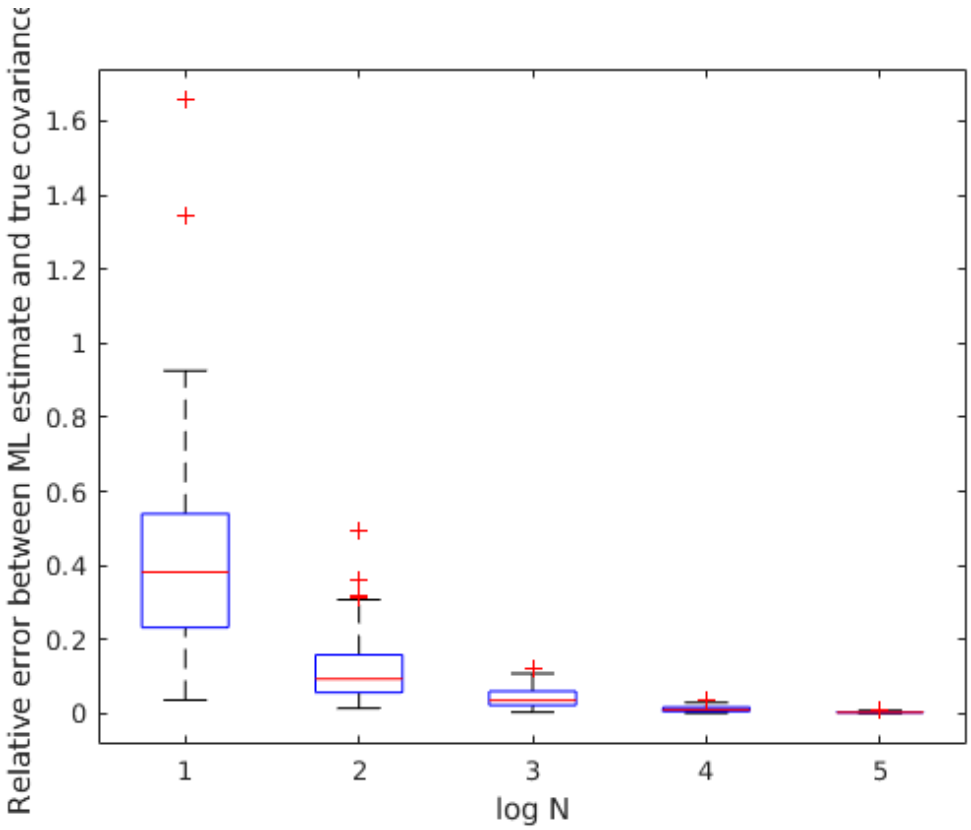
```
rng(5);
C = [1.6250 -1.9486; -1.9486 3.8750];
MU = [1 2]';
[V,D] = eig(C); % get eigenvectors and eigenvalues

A = V*(D^0.5); % as explained in report this is one possible A

ns = [10 10^2 10^3 10^4 10^5];

% the first coordinate represents each trial and the second coordinate
% represents the N we are considering
mean_boxplot_matrix = zeros(100, length(ns));
covariance_boxplot_matrix = zeros(100, length(ns));

for k = 1:length(ns)
   for m = 1:100
       n = ns(k); % current n
       standard_sample = randn(2, n);
       % vectorised sampling, sample is in a 2xN matrix where every
 column
       % is a sample
       sample = MU + A*standard_sample;
       % this gives a 2 x N matrix where every column is a sample

       mean_vector = [0 0]';
       for l=1:n
          mean_vector = mean_vector + sample(:, l);
          % this can be done using sum(.) but we have been instructed
 to
          % use only eig and randn, with sum this is a single line
       end
       mean_vector = mean_vector/n;

       error = norm(mean_vector - MU)/norm(MU);
       mean_boxplot_matrix(m, k) = error;
       % current error value at the mth trial for n

       sample = sample - mean_vector;
       % subtracted mean from sample (to center at origin)
       current_covariance = sample*sample'/n;
       % in the above line we are using the vectorised implementation
 for
       % getting sample covariance. sample is 2xN, sample' is Nx2.
       % Multiplying these two matrices and dividing by N gives the
       % covariance (can be seen by multiplying them explicitly)


       error = norm(C - current_covariance, 'fro')/norm(C, 'fro');
       covariance_boxplot_matrix(m, k) = error;
   end

end
```

```
% below we plot using boxplot
figure;
axis equal;
boxplot(mean_boxplot_matrix);
xlabel("log N");
ylabel("Relative error between ML estimate and true mean");

figure;
axis equal;
boxplot(covariance_boxplot_matrix);
xlabel("log N");
ylabel("Relative error between ML estimate and true covariance");
```





Published with MATLAB® R2020a