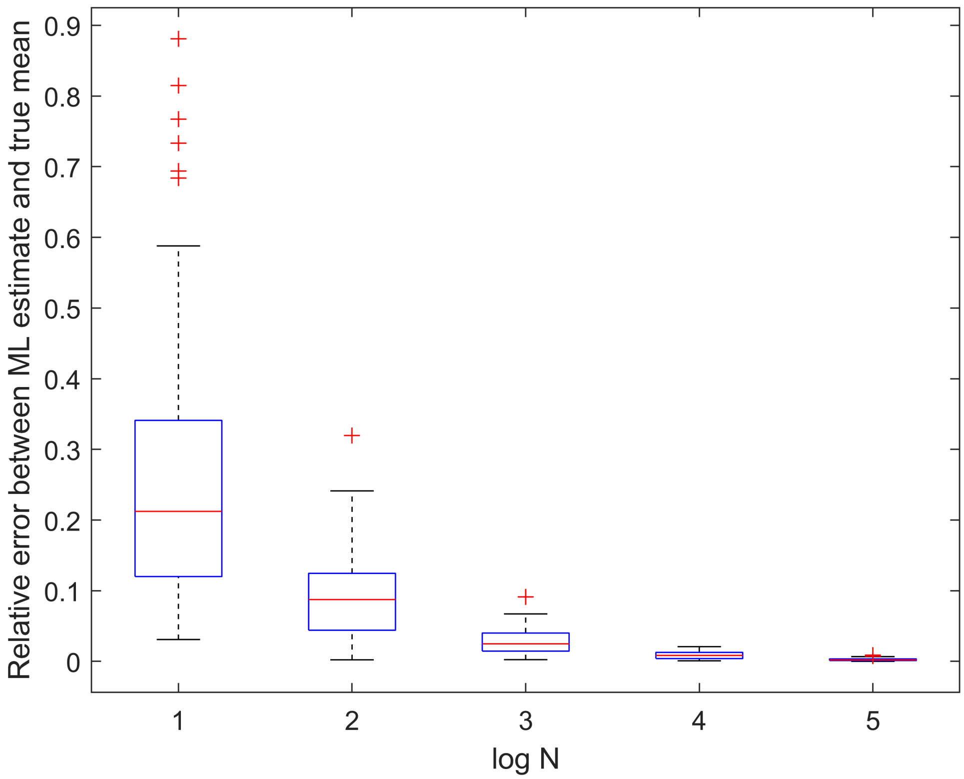
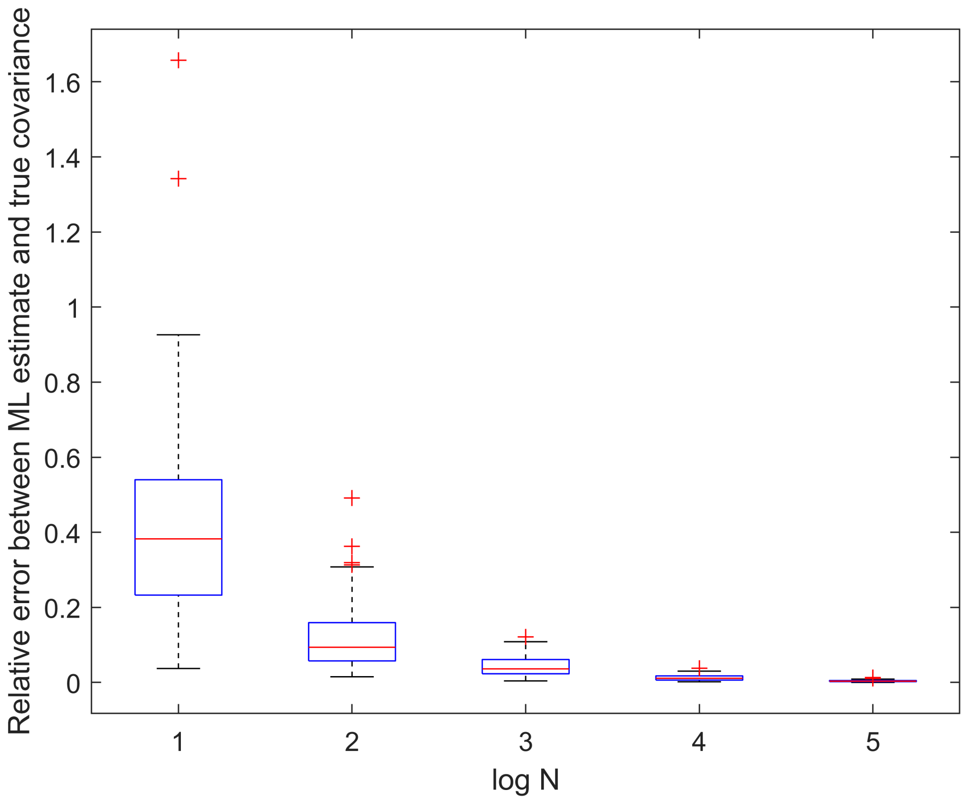
Box Plot for error between true covariance/mean and their MLE estimates

```
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 considerinf
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 sam

       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 out on paper)


       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");
```



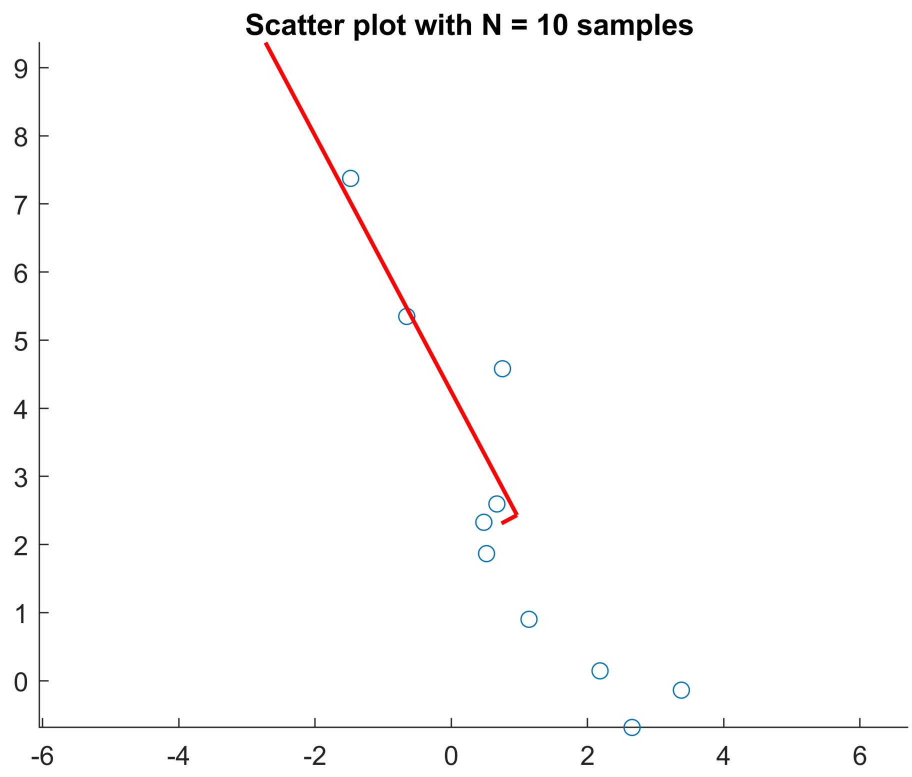

```
figure;
axis equal;
boxplot(covariance_boxplot_matrix);
xlabel("log N");
ylabel("Relative error between ML estimate and true covariance");
```

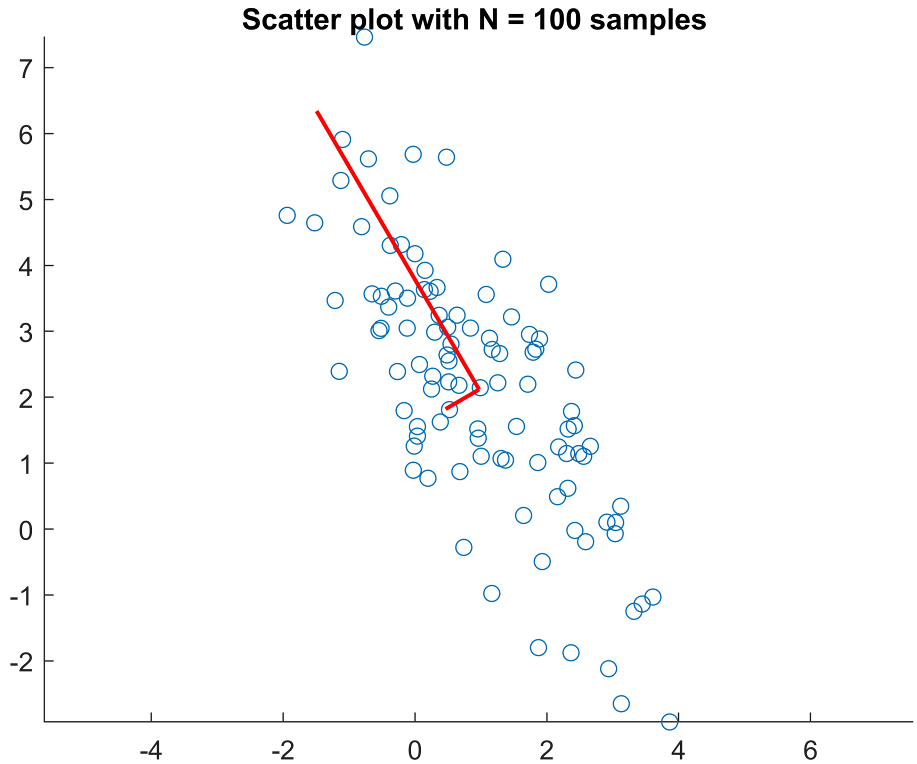


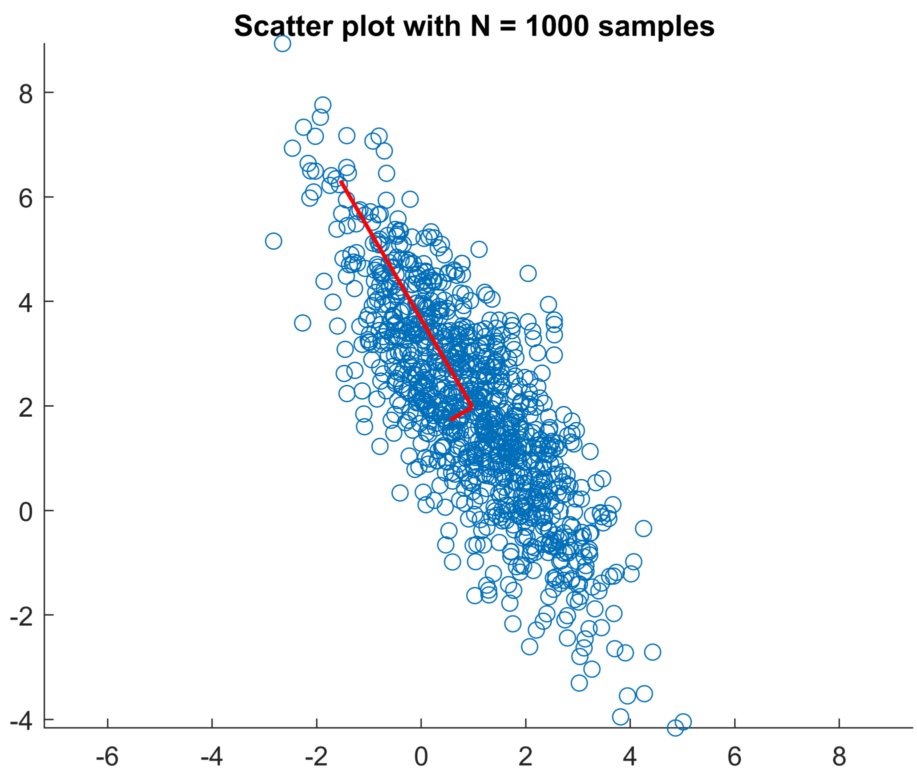
Scatter Plot of Generated Data

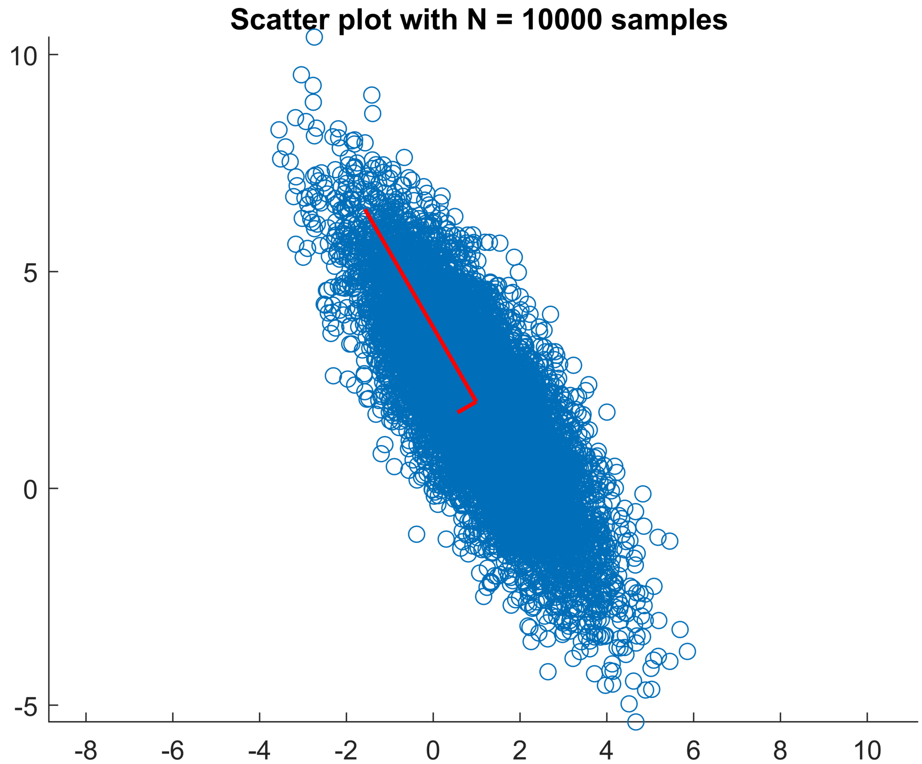
```
rng(5);
C = [1.6250 -1.9486; -1.9486 3.8750];
MU = [1 2]';

[V,D] = eig(C);

A = V*(D^0.5);

ns = [10 10^2 10^3 10^4 10^5];
%same as q2bc.m
for k = 1:length(ns)
    n = ns(k);
    standard_sample = randn(2, n); % standard sample is 2xN matrix where every column is a samp
    sample = MU + A*standard_sample;
    % above transformation gives a 2xN matrix where every column
    % is  a sample from our desired multivariate gaussian
    % Note that MU is getting broadcasted.

    mean_vector = [0 0]';
    for l=1:n
       mean_vector = mean_vector + sample(:, l); % mean can be found using sum
       % but we have been asked to use eig and randn only
    end
    mean_vector = mean_vector/n;
```

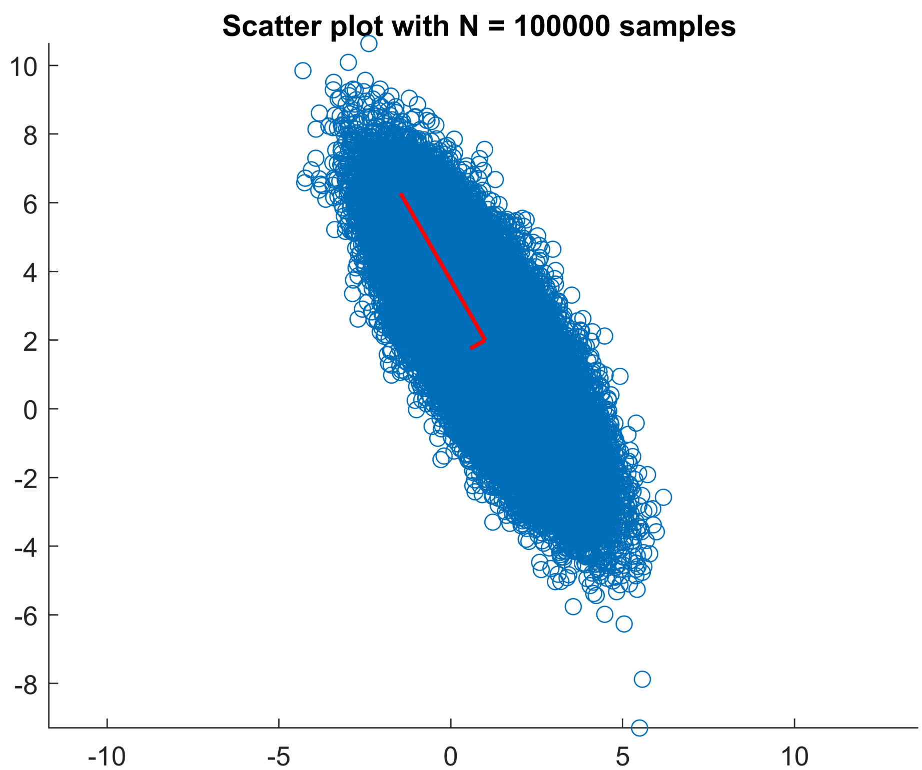
```
    sample = sample - mean_vector; % mean subtraction from sample to center
    current_covariance = sample*sample'/n;
    % vectorised implementation for getting sample covariance

    [v,d] = eig(current_covariance); % this gives the eigenvalues and eigenvectors
    % of the sample covariance which we will need to draw the directions of
    % maximal variance (along the eigenvector)

    figure;

    sample = sample + mean_vector;
    % adding back the mean_vector to get the original sample back

    scatter(sample(1, :), sample(2, :)); % scatter plot
    title("Scatter plot with N = " + string(n) + " samples ");
    axis equal;

    terminal_one = mean_vector + d(1,1)*v(:, 1); % this is the end point
    % of the first line that we will draw, along the first eigenvector
    % with length equal to first eigenvalue
    terminal_two = mean_vector + d(2,2)*v(:, 2);
    % same for the second line

    % below commands draw the lines
    line([mean_vector(1) terminal_one(1)], [mean_vector(2) terminal_one(2)], 'Color', 'r', 'Lir
    line([mean_vector(1) terminal_two(1)], [mean_vector(2) terminal_two(2)], 'Color', 'r', 'Lir

end
```




Scatter plot with N = 100 samples
7
6
5
4
3
2
1
0
-1
-2
-4
-2
0
2
4
6


Scatter plot with N = 1000 samples
8
6
4
2
0
-2
-4
-6
-4
-2
0
2
4
6
8


Scatter plot with N = 10000 samples
10
5
0
-5
-8
-6
-4
-2
0
2
4
6
8
10


Scatter plot with N = 100000 samples
10
8
6
4
2
0
-2
-4
-6
-8
-10
-5
0
5
10