

# ELEN E6880: RMT with Applications

## Homework #0

Chenye Yang cy2540@columbia.edu

September 26, 2019

### P1

#### (a)

Code:

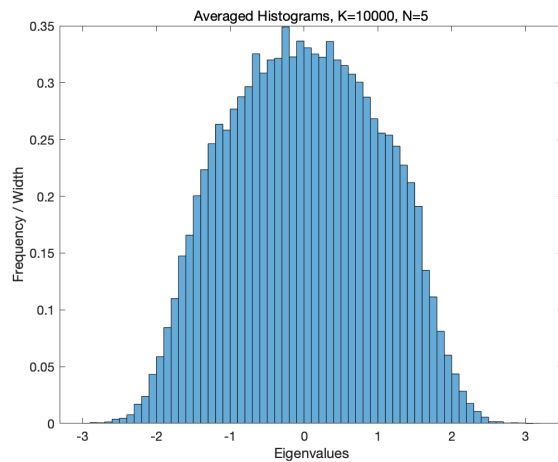
```
1 clear; clc;
2 K = 10000;
3 N = 20;
4
5 mu = 0;           % mean value
6 var = 1/N;        % variance
7 sd = sqrt(var);   % standard deviation
8 A = cell(1, K);
9
10 % Creat an ensemble of K realizations of N x N Gaussian symmetric
    matrices
11 for i = 1:1:K      % index to an ensemble of K
    realizations
12     for j = 1:1:N   % index to row of one matrix
13         for k = j:1:N % index to column of one matrix
14             A{i}(j,k) = normrnd(mu,sd);
15             A{i}(k,j) = A{i}(j,k); % make the matrix symmetric
16         end
17     end
18 end
19
20 e = zeros(N,K);
21 for i = 1:1:K
22     e(:, i) = eig(A{i});
23 end
24
25 % Plot a histogram with Normalization set to 'pdf' to produce an
    estimation of the probability density function.
26 histogram(e, 'Normalization','pdf')
```

```

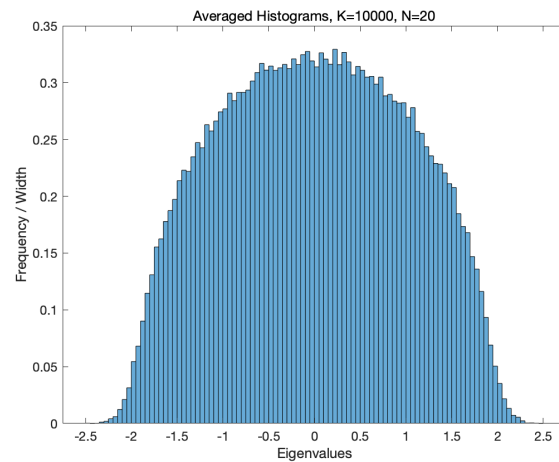
27 ylabel( 'Frequency / Width' );
28 xlabel( 'Eigenvalues' );
29 title( 'Averaged Histograms , K=10000 , N=20' );
30 saveas( gcf, '/Users/yangchenye/Downloads/HW0_1_a_10000_20.png' )

```

Result:



(a)  $K = 10000, N = 5$



(b)  $K = 10000, N = 20$

Figure 1: Averaged histograms of the eigenvalues of A

(b)

Code:

```

1 clear; clc;
2 K = 1;
3 N = 4000;
4
5 mu = 0;           % mean value
6 var = 1/N;        % variance
7 sd = sqrt(var);   % standard deviation
8 A = cell(1, K);
9
10 % Creat an ensemble of K realizations of N x N Gaussian symmetric
    matrices
11 for i = 1:1:K           % index to an ensemble of K
    realizations
12     for j = 1:1:N        % index to row of one matrix
13         for k = j:1:N    % index to column of one matrix
14             A{i}(j,k) = normrnd(mu,sd);
15             A{i}(k,j) = A{i}(j,k); % make the matrix symmetric
16         end
17     end

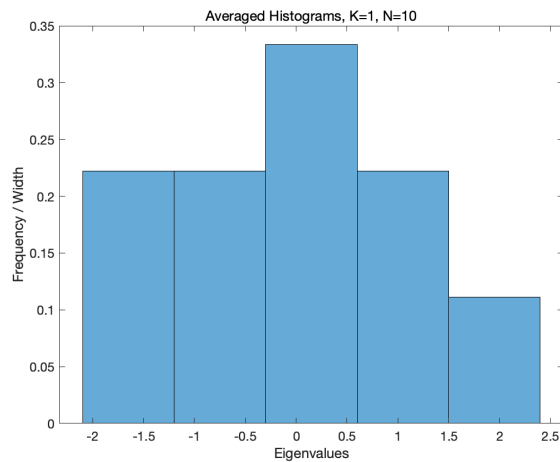
```

```

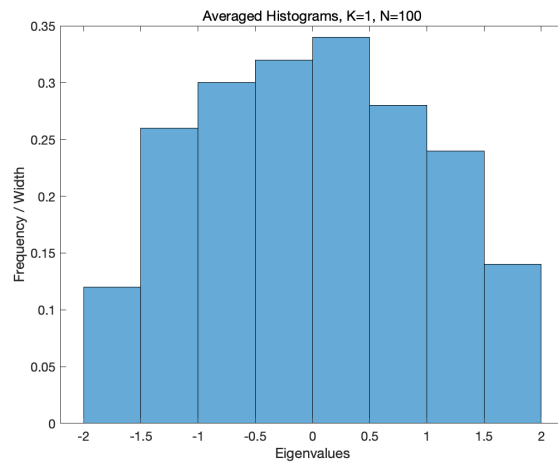
18 end
19
20 e = zeros(N,K);
21 for i = 1:1:K
22     e(:, i) = eig(A{i});
23 end
24
25 % Plot a histogram with Normalization set to 'pdf' to produce an
    estimation of the probability density function.
26 histogram(e, 'Normalization', 'pdf')
27 ylabel('Frequency / Width');
28 xlabel('Eigenvalues');
29 title('Averaged Histograms, K=1, N=4000');
30 saveas(gcf, '/Users/yangchenye/Downloads/HW0_1_b_1_4000.png')

```

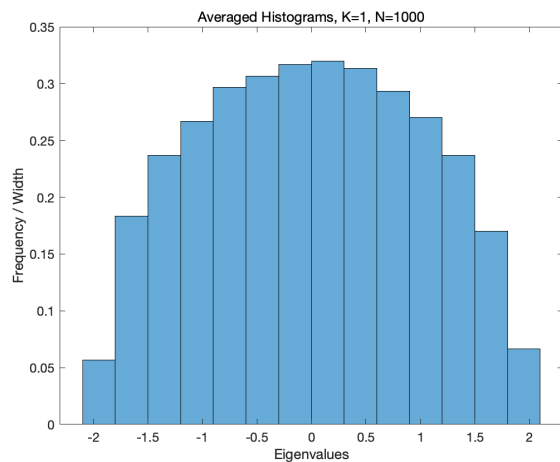
**Result:**



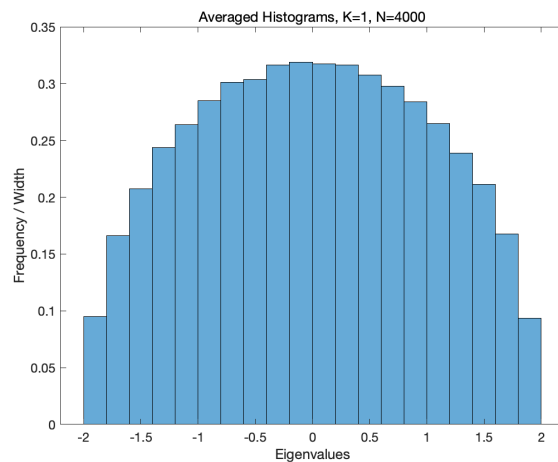
(a)  $K = 1, N = 10$



(b)  $K = 1, N = 100$



(c)  $K = 1, N = 1000$



(d)  $K = 1, N = 4000$

Figure 2: Averaged histograms of the eigenvalues of  $A$

(c)

Code:

```

1 function [p] = Wigner(lamda)
2 % the analytical curve (Wigner's semi-circle law of the limiting
   distribution)
3 for l = 1:length(lamda)
4     p(l) = (1/(2*pi))*sqrt(4-lamda(l)^2)*(abs(lamda(l))<2);
5 end

1 % a
2 clear;clc;
3 KK1 = [10000];
4 NN1 = [5, 20];
5
6 for kindex = 1:length(KK1)
7     for nindex = 1:length(NN1)
8
9         K = KK1(kindex);
10        N = NN1(nindex);
11
12        mu = 0;           % mean value
13        var = 1/N;        % variance
14        sd = sqrt(var); % standard deviation
15        A = cell(1, K);
16
17        % Creat an ensemble of K realizations of N x N Gaussian
           symmetric matrices
18        for i = 1:1:K           % index to an ensemble of
           K realizations
19            for j = 1:1:N       % index to row of one
           matrix
20                for k = j:1:N % index to column of one
           matrix
21                    A{i}(j,k) = normrnd(mu,sd);
22                    A{i}(k,j) = A{i}(j,k); % make the matrix
           symmetric
23                end
24            end
25        end
26
27        e = zeros(N,K);
28        for i = 1:1:K
29            e(:, i) = eig(A{i});
30        end
31        % Plot a histogram with Normalization set to 'pdf' to produce
           an estimation of the probability density function.

```

```

32     histogram(e, 'Normalization', 'pdf')
33     ylabel('p(\lambda)');
34     xlabel('\lambda');
35     title(['Averaged Histograms, K=', num2str(K), ', N=', num2str(N)
36           ']);
37     hold on
38     plot([-3:0.1:3], Wigner([-3:0.1:3]), 'r', 'linewidth', 2);
39     legend('histograms', 'analytical_curve');
40     saveas(gcf, ['/Users/yangchenye/Downloads/HW0_1_c_', num2str(K),
41               '_ ', num2str(N), '.png'])
42     close;
43 end
44
45 % b
46 clear; clc;
47 KK2 = [1];
48 NN2 = [10, 100, 1000, 4000];
49
50 for kindex = 1:length(KK2)
51     for nindex = 1:length(NN2)
52
53         K = KK2(kindex);
54         N = NN2(nindex);
55
56         mu = 0;           % mean value
57         var = 1/N;        % variance
58         sd = sqrt(var);   % standard deviation
59         A = cell(1, K);
60
61         % Creat an ensemble of K realizations of N x N Gaussian
62         % symmetric matrices
63         for i = 1:1:K      % index to an ensemble of
64                             % K realizations
65             for j = 1:1:N  % index to row of one
66                             % matrix
67                 for k = j:1:N % index to column of one
68                             % matrix
69                     A{i}(j,k) = normrnd(mu, sd);
70                     A{i}(k,j) = A{i}(j,k); % make the matrix
71                                     symmetric
72                 end
73             end
74         end
75
76         e = zeros(N,K);

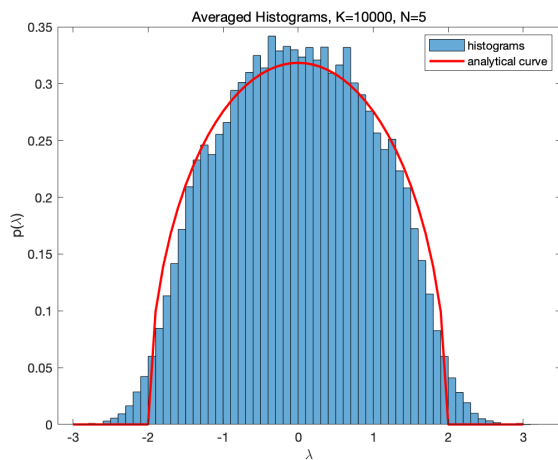
```

```

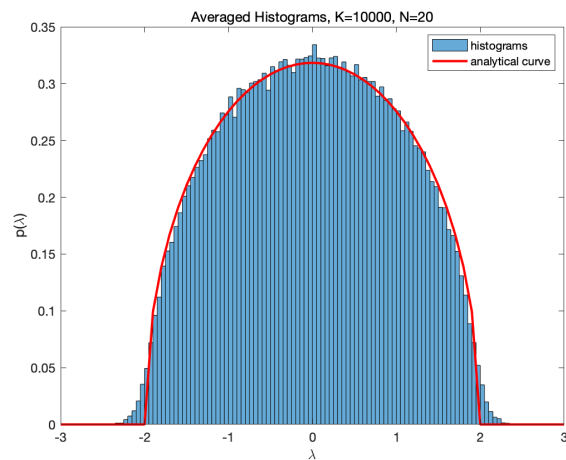
72     for i = 1:1:K
73         e(:, i) = eig(A{i});
74     end
75     % Plot a histogram with Normalization set to 'pdf' to produce
       an estimation of the probability density function.
76     histogram(e, 'Normalization', 'pdf')
77     ylabel('p(\lambda)');
78     xlabel('\lambda');
79     title(['Averaged Histograms, K=', num2str(K), ', N=', num2str(N)
           ]);
80     hold on
81     plot([-3:0.1:3], Wigner([-3:0.1:3]), 'r', 'linewidth', 2);
82     legend('histograms', 'analytical curve');
83     saveas(gcf, ['/Users/yangchenye/Downloads/HW0_1_c_', num2str(K),
           '_ ', num2str(N), '.png'])
84     close;
85 end
86 end

```

Result:



(a)  $K = 10000, N = 5$



(b)  $K = 10000, N = 20$

Figure 3: Averaged histograms and analytical curve, convergence (a)

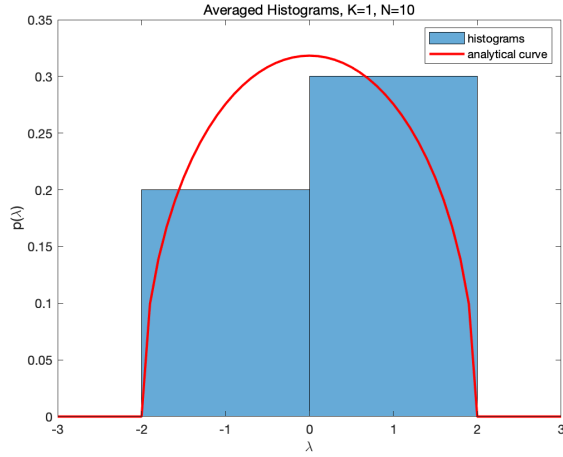
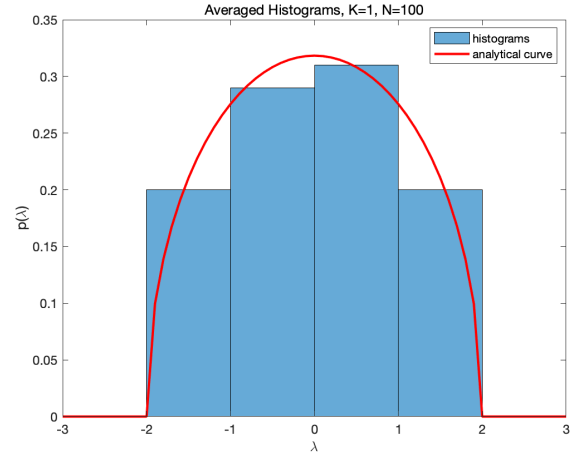
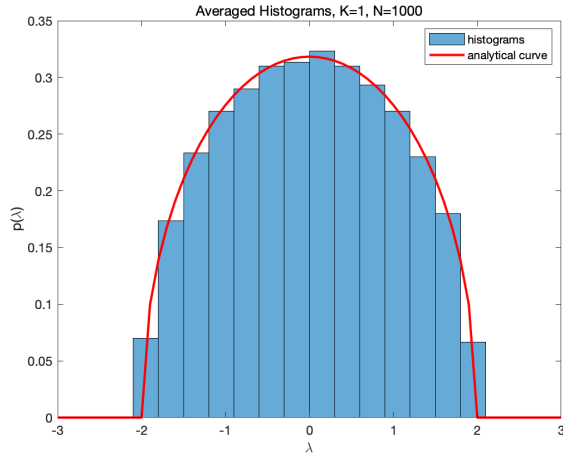
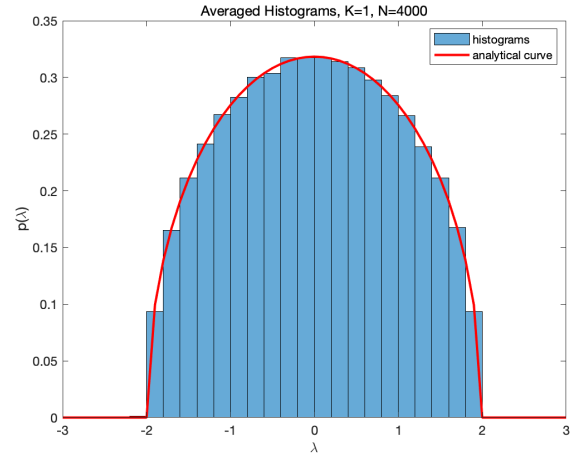
(a)  $K = 1, N = 10$ (b)  $K = 1, N = 100$ (c)  $K = 1, N = 1000$ (d)  $K = 1, N = 4000$ 

Figure 4: Averaged histograms and analytical curve, convergence (b)

(d)

**Ans:**

The convergence in P2(a) is through increasing the number of different matrices, while the convergence in P2(b) is through increasing the dimension of one matrix.

From Figure 3 and Figure 4, it is obvious that the frequency histograms will converge to analytical curve with the increase of number of eigenvalues. This convergence is like Bootstrap Distribution, using the histogram of many samples to estimate a real distribution.

In Figure 3(a) and Figure 3(b), because the number of matrices is very large and the dimension is small, there exists many eigenvalues belonging to interval  $[-3, -2]$  and  $[2, 3]$ . But the analytical curve is 0 in those intervals. The convergence is not exactly fitting the analytical curve and has some overshoot outside interval  $(-2, 2)$ .

In Figure 4(c) and Figure 4(d), because the dimension of matrix is extremely large, the frequency of eigenvalues falling outside interval  $(-2, 2)$  is very small. Thus, the convergence will exactly fit the analytical curve.

Also, we can find, although with a smaller amount of eigenvalues, the convergence in P1(b) is

better than that in P1(a).

## P2

(a)

Result:

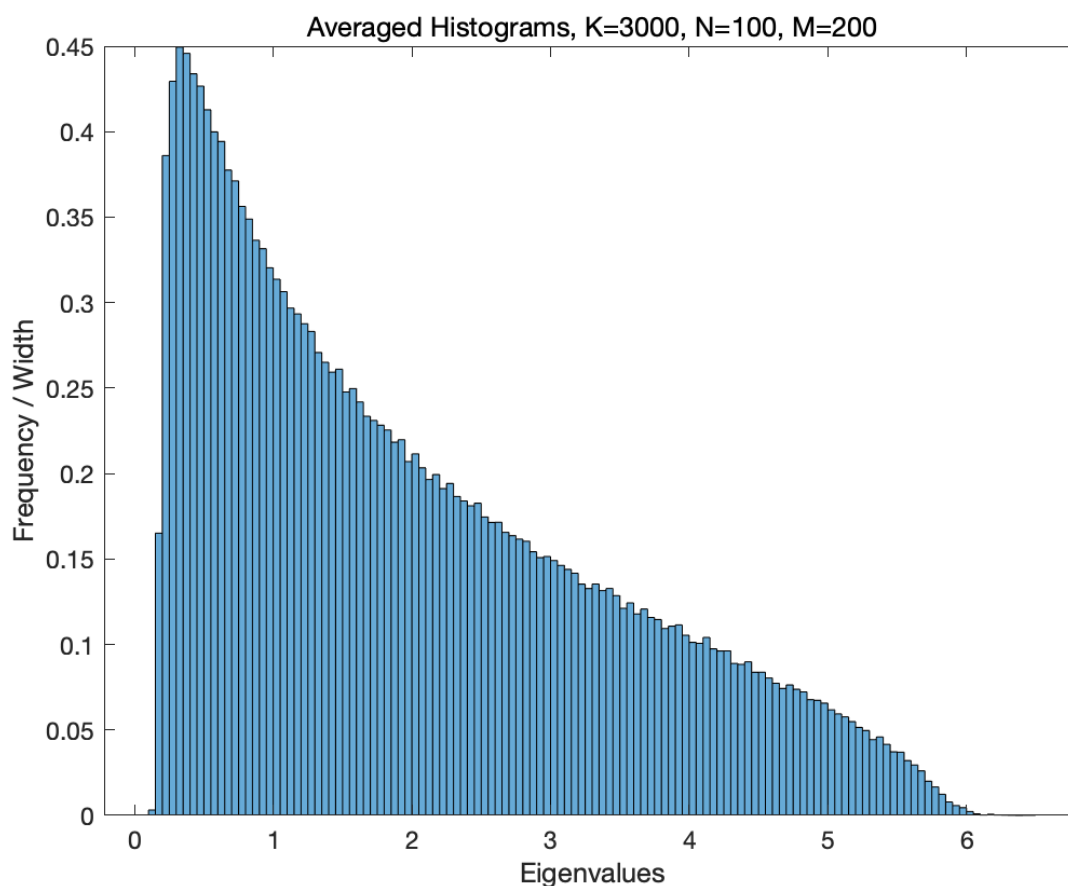


Figure 5: Averaged histograms of the eigenvalues of A

Code:

```

1 K = 3000;
2 N = 100;
3 M = 2*N;
4
5 % Creat an ensemble of K realizations of N x N Wishart matrices
6 A = cell(1,K);
7 for i = 1:1:K
8     X = randn(N,M);
9     A{i} = (1/N)*(X*X');

```



```

10 end
11
12 e = zeros(N,K);
13 for i = 1:1:K
14     e(:, i) = eig(A{i});
15 end
16
17 % Plot a histogram with Normalization set to 'pdf' to produce an
    estimation of the probability density function.
18 histogram(e, 'Normalization','pdf')
19 ylabel('Frequency_/Width');
20 xlabel('Eigenvalues');
21 title(['Averaged_Histograms , K=', num2str(K), ', N=', num2str(N), ', M=',
    num2str(M)]);
22 hold on
23 saveas(gcf, [' /Users/yangchenye/Downloads/HW0_2_a_', num2str(K), '_ ',
    num2str(N), '_ ', num2str(M), '.png'])

```

(b)

Code:

```

1 K = 1;
2 N = 2000;
3 M = 2*N;
4
5 % Creat an ensemble of K realizations of N x N Wishart matrices
6 A = cell(1,K);
7 for i = 1:1:K
8     X = randn(N,M);
9     A{i} = (1/N)*(X*X');
10 end
11
12 e = zeros(N,K);
13 for i = 1:1:K
14     e(:, i) = eig(A{i});
15 end
16
17 % Plot a histogram with Normalization set to 'pdf' to produce an
    estimation of the probability density function.
18 histogram(e, 'Normalization','pdf')
19 ylabel('Frequency_/Width');
20 xlabel('Eigenvalues');
21 title(['Averaged_Histograms , K=', num2str(K), ', N=', num2str(N), ', M=',
    num2str(M)]);
22 hold on

```

```
23 saveas(gcf,['/Users/yangchenye/Downloads/HW0_2_b_',num2str(K),'_',
    num2str(N),'_',num2str(M),'_png'])
```

**Result:**

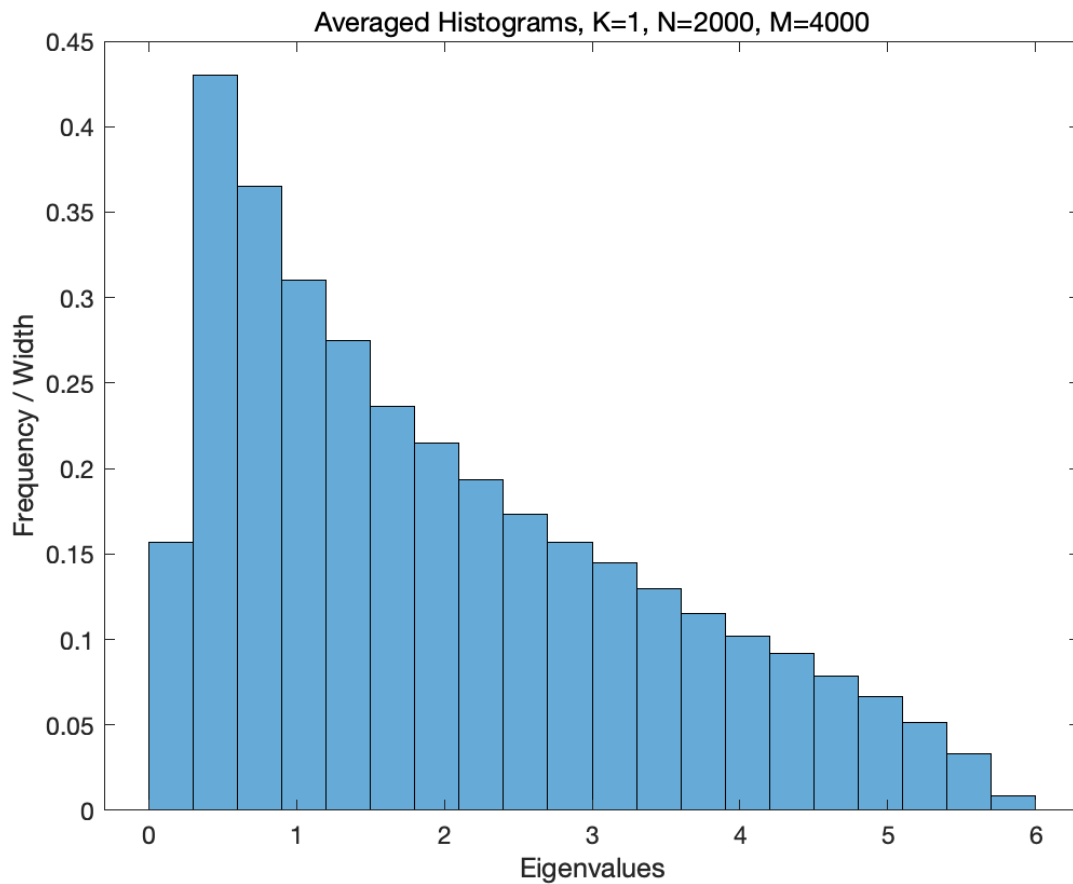


Figure 6: Averaged histograms of the eigenvalues of A

**(c)**

**Code:**

```
1 function [p] = MarcenkoPastur(lamda,N,M)
2 % Marcenko-Pastur law of the limiting distribution
3
4 alpha = M./N;
5 for i = 1:length(lamda)
6     p(i) = sqrt(4*alpha-(lamda(i)-1-alpha).^2)./(2*pi*lamda(i))*
7         logical((1-sqrt(alpha)).^2 <= lamda(i)) && logical(lamda(i) <=
8             (1+sqrt(alpha)).^2));
9 end
10 end
```

```

1 KK = [3000, 1];
2 NN = [100, 2000];
3 MM = [2*NN(1), 2*NN(2)];
4
5 for question = 1:2
6     K = KK(question);
7     N = NN(question);
8     M = MM(question);
9     % Creat an ensemble of K realizations of N x N Wishart matrices
10    A = cell(1,K);
11    for i = 1:1:K
12        X = randn(N,M);
13        A{i} = (1/N)*(X*X');
14    end
15
16    e = zeros(N,K);
17    for i = 1:1:K
18        e(:, i) = eig(A{i});
19    end
20
21    % Plot a histogram with Normalization set to 'pdf' to produce an
22    % estimation of the probability density function.
23    histogram(e, 'Normalization','pdf')
24    ylabel('p(\lambda)');
25    xlabel('\lambda');
26    title(['Averaged Histograms, K=', num2str(K), ', N=', num2str(N), ', M',
27    '= ', num2str(M)]);
28    hold on
29    plot([0.1:0.1:6], MarcenkoPastur([0.1:0.1:6],N,M), '-r', 'LineWidth', 2);
30    legend('histograms','analytical_curve');
31    saveas(gcf,['/Users/yangchenye/Downloads/HW0_2_c_', num2str(K), '_ ',
32    num2str(N), '_ ', num2str(M), '.png'])
33    close;
34 end

```

Result:

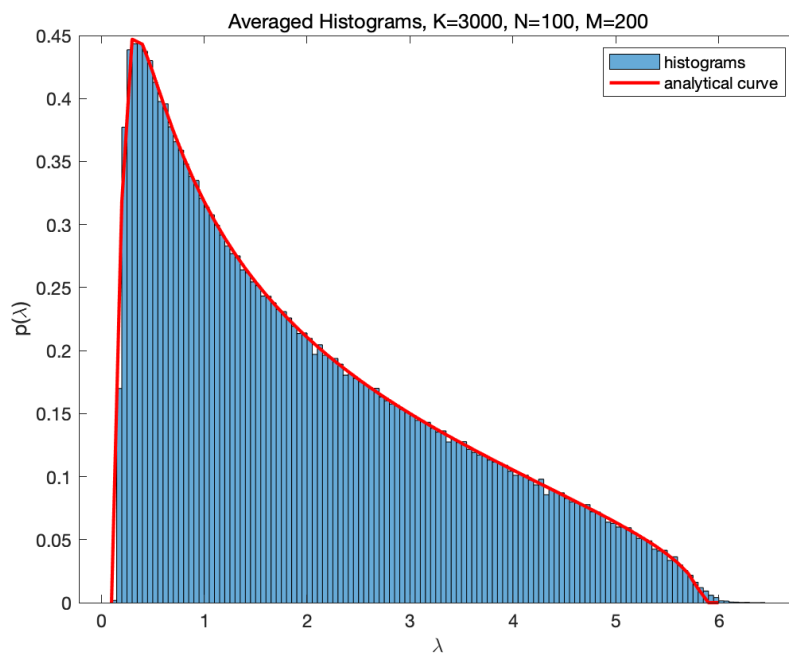


Figure 7: Averaged histograms and analytical curve, convergence (a)

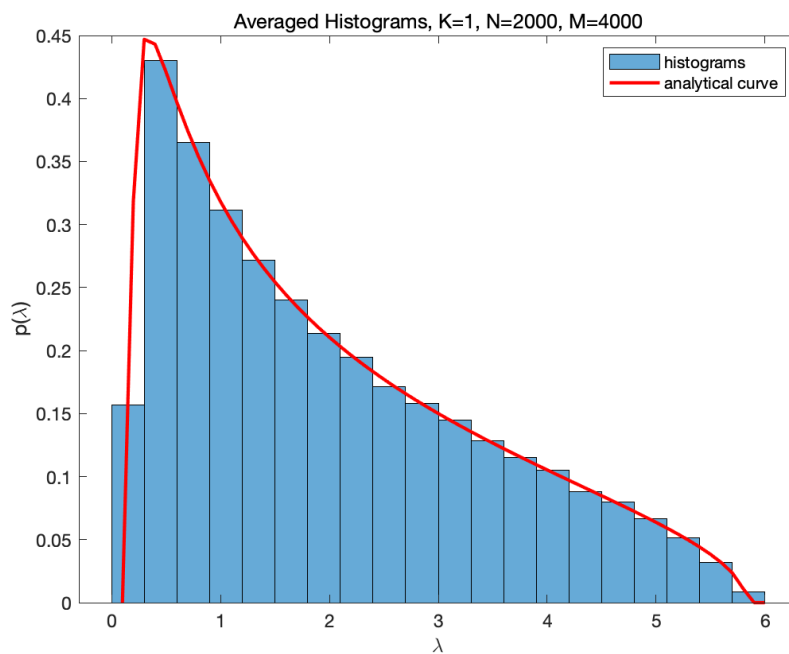


Figure 8: Averaged histograms and analytical curve, convergence (b)

(d)

**Ans:**

Firstly, it should be notified that the seemingly "difference" between convergence effect is only caused by the bin width of histogram. Plot two histograms with same bin width in one figure below (Figure 9):

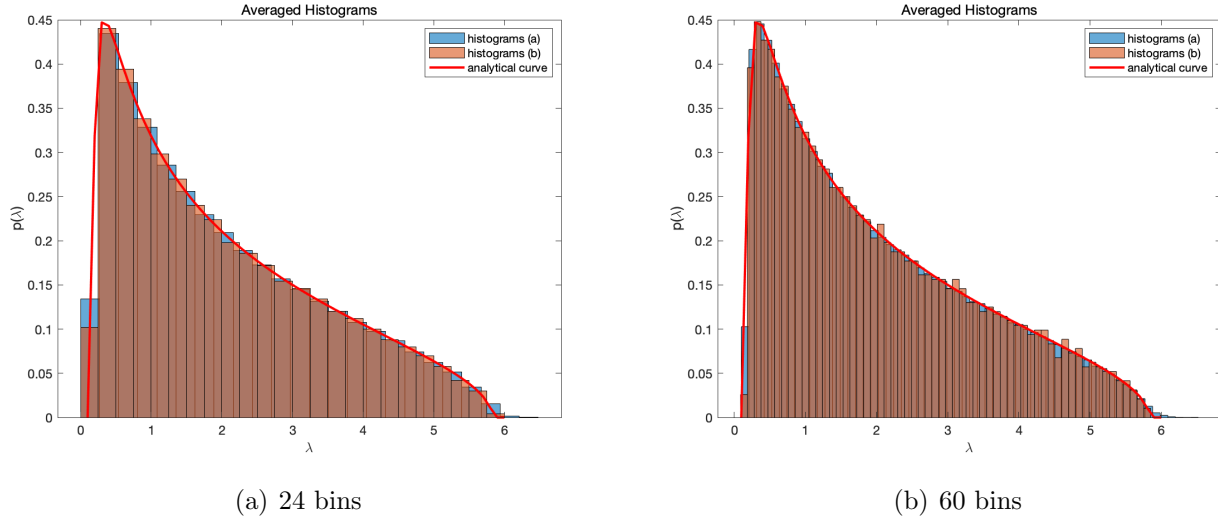


Figure 9: Different bin width

Similarly with Problem 1, the convergence in P2(a) is through increasing the number of different matrices, while the convergence in P2(b) is through increasing the dimension of one matrix.

From Figure 7 and Figure 8, it is obvious that the frequency histograms will converge to analytical curve with the increase of number of eigenvalues. This convergence is like Bootstrap Distribution, using the histogram of many samples to estimate a real distribution.

In Figure 7, because the number of matrices is very large and the dimension is small, there exists many eigenvalues belonging to interval  $U(6, \delta) = \{x \mid 6 - \delta < x < 6 + \delta\}$ . But the analytical curve is 0 in those intervals. The convergence is not exactly fitting the analytical curve and has some overshoot outside interval  $(0, 6)$

In Figure 8, because the dimension of matrix is extremely large, the frequency of eigenvalues falling outside interval  $(0, 6)$  is very small. Thus, the convergence will exactly fit the analytical curve.

Also, from Figure 9, we can have a clear view that histogram in P2(b) (yellow) has a better convergence than that in P2(a) (blue), even though the total number of eigenvalues in P2(b) is smaller.