Group Simulation Activity 2

Emma Chan (20206641)

Charlotte Lombard (20232888)

Jake Moffat (20212243)

Jack Mason (20060348)

Question 1

Part I

The joint probability mass function (PMF) of discrete random variables **x** and **y** as a function of x and y was generated in the form of a three-dimensional stem plot. This is displayed below in Figure 1.

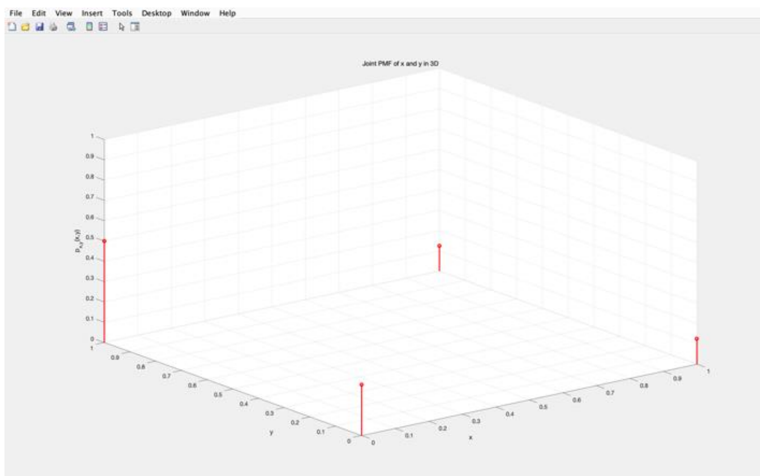

*Figure 1: Joint PMF of **x** and **y***

Next, the joint cumulative distribution function (CDF) of random variables **x** and **y** was computed through the summation of the values of the joint PMF. This was also plotted as a three-dimensional stem plot, seen below in Figure 2.

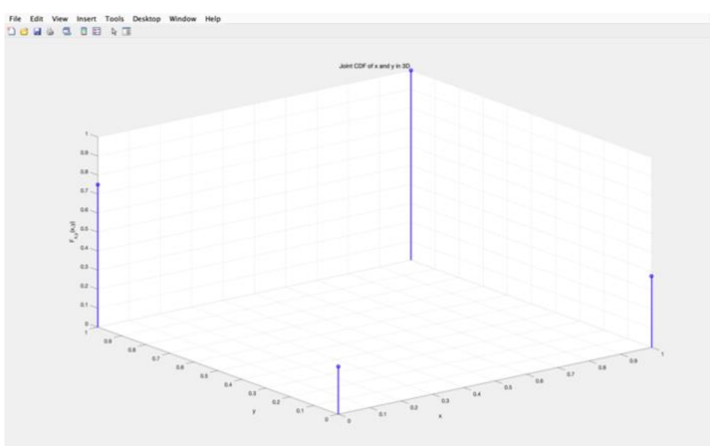

*Figure 2: Joint CDF of **x** and **y***

The code written in MATLAB for this part of the problem is displayed in Figure 3 below.

```matlab
% ELEC 326
% Group Simulation Activity 2
% Question 1 - Part I
% Emma Chan, Charlotte Lombard, Jack Mason, Jake Moffat

% Prompt: Write a MATLAB code to plot the joint PMF and joint CDF of
% these two RVs in a 3D space (z axis shows the joint probability value).

% Load the RVs - don't need for this part
% RV1 = load('RV1.mat').RV1;
% RV2 = load('RV2.mat').RV2;
% RV3 = load('RV3.mat').RV3;

% Initialize x and y values in the grid
[x, y] = meshgrid(0:1, 0:1);

% define the joint PMF of x and y
jointPMF = [0.25, 0.125; 0.5, 0.125];

% define the joint CDF of x and y
jointCDF = [jointPMF(1, 1), jointPMF(1, 1) + jointPMF(1, 2); jointPMF(1, 1) + jointPMF(2, 1), sum(jointPMF, 'all')];

% plot the joint PMF in 3D
plot1 = figure('Name', 'Joint PMF');
%stem3(x, y, jointPMF, 'Color', 'r');
%stem3(x, y, jointPMF, 'LineWidth', 2);
stem3(x, y, jointPMF, 'Color', 'r', 'LineWidth', 2);

title('Joint PMF of x and y in 3D', 'FontWeight', 'normal');
set(gca, 'FontSize', 11);
axis1 = gca;
axis1.XLabel.String = 'x';
% axis1.XLim = [0, 100];
axis1.YLabel.String = 'y';
axis1.ZLabel.String = 'p_{x,y}(x,y)';
axis1.ZLim = [0, 1];

% plot the joint CDF in 3D
plot2 = figure('Name', 'Joint CDF');
% stem3(x, y, jointCDF, 'Color', 'b');
% stem3(x, y, jointCDF, 'LineWidth', 2);
stem3(x, y, jointCDF, 'Color', 'b', 'LineWidth', 2);

title('Joint CDF of x and y in 3D', 'FontWeight', 'normal');
set(gca, 'FontSize', 11);
axis2 = gca;
axis2.XLabel.String = 'x';
% axis2.XLim = [0, 100];
axis2.YLabel.String = 'y';
axis2.ZLabel.String = 'F_{x,y}(x,y)';
axis2.ZLim = [0, 1];
```

*Figure 3: MATLAB code used to compute and plot the joint PDF and CDF of **x** and **y***

Part II

To generate 100000 realizations of (**x,y**) and store them in the matrix XY, 100000 uniformly distributed random numbers between 0 and 1 were generated. Markers were used to separate these random values to correspond with the given joint pmf of **x** and **y.** The MATLAB script written to complete this is displayed for reference in Figure 4 below.

```matlab
% ELEC 326
% Group Simulation Activity 2
% Question 1 - Part II
% Emma Chan, Charlotte Lombard, Jack Mason, Jake Moffat

% Prompt: Write a MATLAB code to generate N = 100000 realizations of (x,y)
% and save the observed values in a 2×100000 matrix named XY.

% define the number of trials (100000)
N = 1e5;

% generate N = 100000 realizations of (x,y)
numN = rand(1,N);

% initialize matrix named XY
XY = zeros(2, N);

% using the table, create joint PMF cases
% case (0, 0) = 0.0 | case (0, 1) = 0.1 |
% case (1, 0) = 1.0 | case (1, 1) = 1.1 |

% for case (0, 0)
numN(numN <= 0.25) = 0.1;

% case (0, 1)
numN((numN > 0.25) & (numN <= 0.75)) = 0.2;

% case (1, 0)
numN((numN > 0.75) & (numN <= 0.875)) = 0.3;

% case (1, 1)
numN(numN > 0.875) = 0.4;

% save observed values in XY matrix
for n = 1:N
    switch(numN(n))
        case 0.1 % (0, 0)
            XY(:, n) = [0; 0];
        case 0.2 % (0 , 1)
            XY(:, n) = [0; 1];
        case 0.3 % (1 , 0)
            XY(:, n) = [1; 0];
        case 0.4 % (1 , 1)
            XY(:, n) = [1; 1];
    end
end
```

*Figure 4: MATLAB code used to compute 1000000 realizations of (**x,y**)*

Part III

The experimental joint PMF of **x** and **y** was computed using the 2 x 100000 matrix XY that was generated in Part II. This is displayed in the three-dimensional stem plot in Figure 5 below.

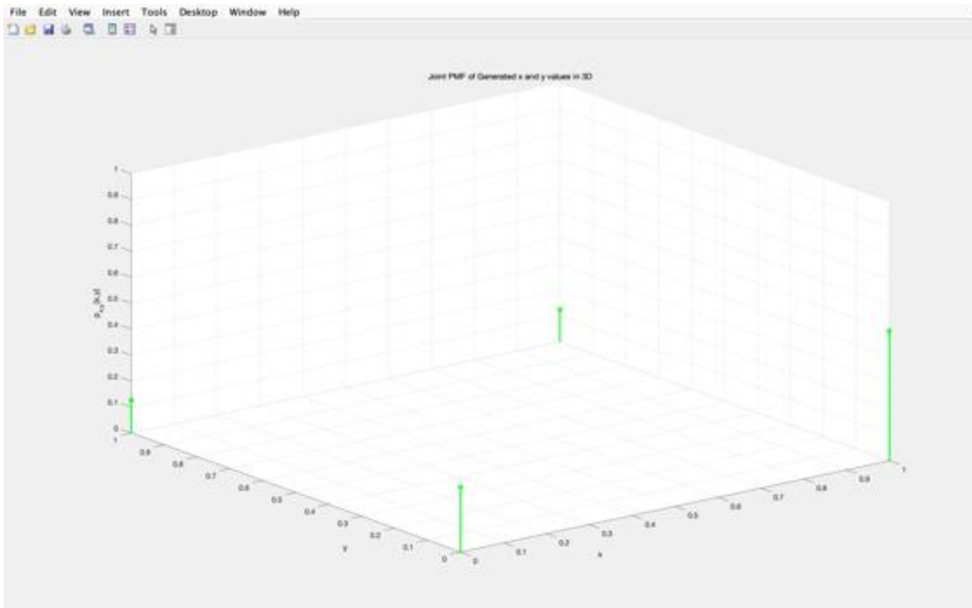

*Figure 5: Experimental joint PMF of **x** and **y***

The appearance of Figure 5 is extremely like that of Figure 1, meaning that the expected and experimental joint PMFs are almost the same, and so the trials were properly conducted. However, there is some difference in the numerical values of the expected and experimental joint PMFs.

*Table 1: Expected joint PMF values of **x** and **y**.*

	y = 0	y = 1
x = 0	0.25	0.5
x = 1	0.125	0.125

*Table 2: Experimental joint PMF values of **x** and **y.***

	y = 0	y = 1
x = 0	0.2504	0.4999
x = 1	0.1242	0.1256

The values seen in Tables 1 and 2 differ slightly by a margin of approximately 1%. This is considered a negligible margin of error. To further improve the difference between the expected and experimental values, the number of trials should be increased. The code for this part of the problem can be found for reference in Figure 6 below. It is a continuation to the code from part II.

```matlab

% -----------------------------------------------------------------------
% Part III
% Prompt: To this end, you need to write a MATLAB code to count the
% number of times each possible outcome is observed and save them in a
% vector named H (for example, if (x = 1,y = 0) is observed 150 times,
% then H(1,0) = 150). Then divide H by the number of trials N and plot
% it in a 3D space and compare it with the joint PMF you calculated in
% section I.

% The Vector named H
H = zeros(2, 2);

% Count the occurences of each case generated in Part II
for n = 1:N;
    numN = XY(:, n);
    numN = numN + 1;
    H(numN(1), numN(2)) = H(numN(1), numN(2)) + 1;
end

% initialzie the Joint PMF to compare to the Joint PMF in Part I
jointPMF = H / N;

% initialize x and y values in the grid
% create the new Joint PMF plot for the generated data in 3D
plot1 = figure('Name', 'Generated Joint PMF');

%stem3(x, y, jointPMF, 'Color', 'r');
%stem3(x, y, jointPMF, 'LineWidth', 2);
stem3(x, y, jointPMF, 'Color', 'g', 'LineWidth', 2);

title('Joint PMF of Generated x and y values in 3D', 'FontWeight', 'normal');
set(gca, 'FontSize', 11);
axis1 = gca;
axis1.XLabel.String = 'x';
% axis1.XLim = [0, 100];
axis1.YLabel.String = 'y';
axis1.ZLabel.String = 'p_{x,y}(x,y)';
axis1.ZLim = [0, 1];

```

*Figure 6: MATLAB code written to plot the experimental joint PMF of **x** and **y**. This code is a continuation of the Part II MATLAB script shown in Figure 4.*

Part IV

The experimental marginal PMFs of **x** and **y** were computed using the joint PMF that was computed in Part III. These values are displayed in Table 3 below.

*Table 3: Experimental joint and marginal PMFs of **x** and **y**.*

	y = 0	*y = 1*	$p_x(x)$
x = 0	0.2504	0.4999	**0.7503**
x = 1	0.1242	0.1256	**0.2497**
$p_y(y)$	**0.3746**	**0.6255**	

The expected marginal PMFs of **x** and **y** were included for reference in Table 4 below.

*Table 4: Expected joint and marginal PMFs of **x** and **y**.*

	y = 0	y = 1	$p_x(x)$
x = 0	0.25	0.5	**0.75**
x = 1	0.125	0.125	**0.25**
$p_y(y)$	**0.375**	**0.625**	

The expected and experimental marginal PMFs seen in the two tables above differ by a margin of at most 0.9%. This is less than the error generated for the experimental joint PMF from Part III. The script written for this part of the problem is displayed in Figure 7 below. It is a continuation to the code from parts II and III.

```
% -------------------------------------------------------------------
% Part IV

% The Vector named H
H = zeros(2, 2);

% Count the occurences of each case generated in Part II
for n = 1:N;
    numN = XY(:, n);
    numN = numN + 1;
    H(numN(1), numN(2)) = H(numN(1), numN(2)) + 1;
end

% initialzie the Joint PMF to compare to the Joint PMF in Part I
jointPMF = H / N;

% Prompt: Write a MATLAB code to estimate the marginal PMFs p_x(x)
% and p_y(y) using the data generated in section II
PMF_x = [sum(jointPMF(1, :)), sum(jointPMF(2, :))];
PMF_y = [sum(jointPMF(:, 1)), sum(jointPMF(:, 2))];
```

*Figure 7: MATLAB script written to compute the experimental marginal PMF of discrete RVs **x** and **y** from their experimental joint PMF. This code is a continuation of the Part II and Part III MATLAB scripts shown in Figures 4 and 6 respectively.*

Question 2

Part I

The experimental joint PMF of RV1 and RV2 was computed using the same method as seen in Part III of Question 1. This was then plotted as a three-dimensional surface displayed in Figure 8 below.

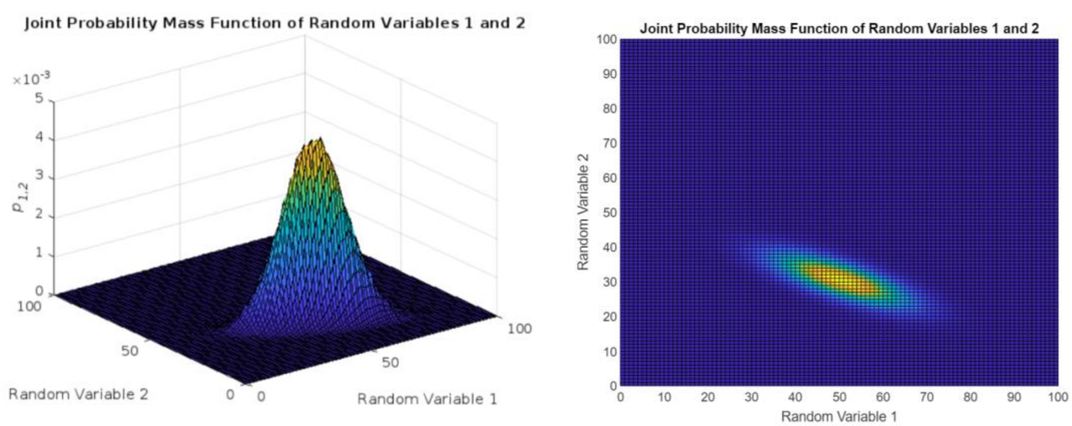

Figure 8: Experimental joint PMF of given discrete random variables 1 and 2

Figure 8 shows that RV1 and RV2 are indeed correlated. The obvious diagonal trend seen in the two-dimensional view of the plot means that each of the random variables has a dependance on the other, and this dependance is due to correlation.

The experimental joint PMF of RV1 and RV3 was also computed using the same method as seen in Part III of Question 1. It was also plotted as a three-dimensional surface and can be seen in Figure 9 below.

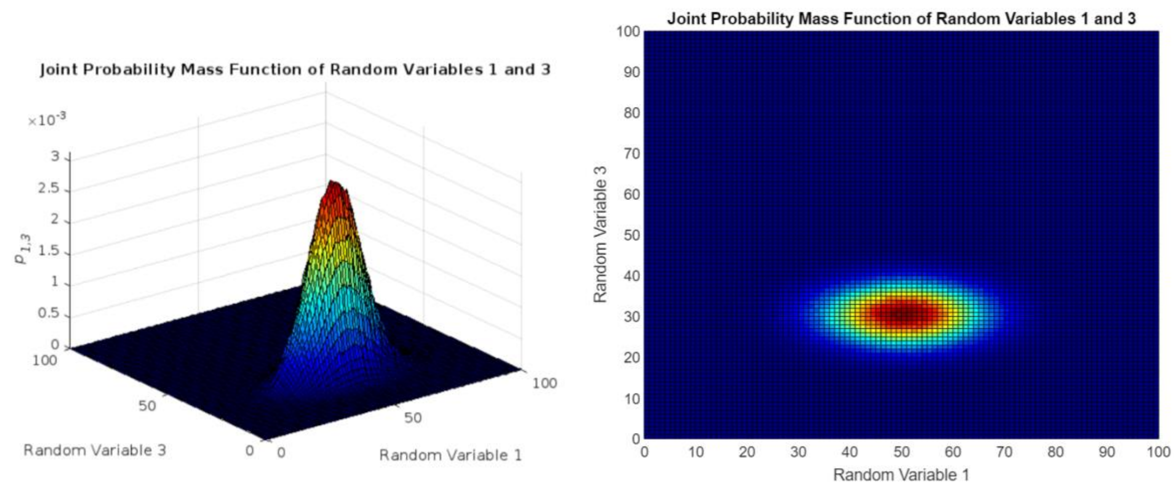

Figure 9: Experimental joint PMF of given discrete variables 1 and 3

The appearance of Figure 9 shows that RV1 and RV3 are uncorrelated. The plot is centered and there is no diagonal trend in the two-dimensional plot which means that the function can be separated into factors that depend on RV1 and RV3 separately. This suggests that RV1 and RV3 are independent of each other, which is another sign that they are uncorrelated.

The MATLAB code written to complete this section of the report is displayed in Figure 10 below.

```
% Initialize the number of values in a random varaible and grid for
% plotting of the graph
N = length(RV1);
[x,y] = meshgrid(0:100, 0:100);

% Compute the joint PMF of values from RV1 and RV2 files
p12 = zeros(101, 101);
for n = 1:N
    i = RV2(n) + 1;
    j = RV1(n) + 1;
    p12(i,j) = p12(i,j) + 1;
end
p12 = p12 / N;

% Generate the 3D plot of the joint PMF for RV1 and RV2 that was computed
% in the last step
figure;
surf(x,y, p12);
xlabel('Random Variable 1');
ylabel('Random Variable 2');
zlabel('{\it p_{1,2}}');
title('Joint Probability Mass Function of Random Variables 1 and 2');

%Compute the joint PMF of values from RV1 and RV3 files
p13 = zeros(101, 101);
for n = 1:N
    i = RV3(n) + 1;
    j = RV1(n) + 1;
    p13(i,j) = p13(i,j) + 1;
end
p13 = p13 / N;

% % Generate the 3D plot of the joint PMF for RV1 and RV3 that was computed
% in the last step
figure;
colormap(jet);
surf(x,y, p13);
xlabel('Random Variable 1');
ylabel('Random Variable 3');
zlabel('{\it p_{1,3}}');
title('Joint Probability Mass Function of Random Variables 1 and 3');
```

Figure 10: MATLAB script written to compute and plot the experimental joint PMF of given discrete RVs 1 and 2, then 1 and 3 respectively

Part II

To justify the answers given in Part I, the estimated correlation coefficient between RV1 and Rv2 and between RV1 and RV3 must be computed. This computation was done using the expression below:

$$\rho_{n,m} = \frac{cov[n,m]}{\sigma_n \sigma_m},$$

where n, m denote the numbers of the given discrete random variable that was used.

The standard deviations of each random variable were computed by using their respective estimated covariances and square rooting them. The covariances were computed using the expression below:

$$cov[n,\ m] = \mathrm{E}[nm] - \mathrm{E}[n]\mathrm{E}[m],$$

where E represents the expected values.

Following the above computation, the correlation coefficients between RV1 and RV2, and then between RV1 and RV3 were determined respectively as shown below:

Table 5: Correlation coefficients between RV1 and RV2 and then between RV1 and RV3.

rho12	-0.7492
rho13	-4.5543e-04

Where rho12 represents the value for the correlation coefficients between random variables 1 and 2, then rho13 represents the value for the correlation coefficient between random variables 1 and 3. As expected from the results in Part I, RV1 and RV2 are correlated from their non-zero correlation coefficient. It is also clear that RV1 and RV3 are approximately uncorrelated as their correlation coefficient is near the ideal 0.

The MATLAB script written to generate these results is displayed for reference below:

```
% Load given random variables
N = length(RV1);

%Estimate the means of each random variable
mu1 = sum(RV1) / N;
mu2 = sum(RV2) / N;
mu3 = sum(RV3) / N;

%Estimate the standard deviations for each random variable
sig1 = sqrt(sum((RV1 - mu1) .* (RV1 - mu1)) /N);
sig2 = sqrt(sum((RV2 - mu2) .* (RV2 - mu2)) /N);
sig3 = sqrt(sum((RV3 - mu3) .* (RV3 - mu3)) /N);

%Compute the joint PMF of RV1 and RV2
p12 = zeros(101, 101);
for n = 1:N
    i = RV2(n) + 1;
    j = RV1(n) + 1;
    p12(i,j) = p12(i,j) + 1;
end
p12 = p12 / N;

%Estimate the expected value of RV1 RV2
mu12 = 0;
for n = 0:100
    for m = 0:100
        mu12 = mu12 + n * m * p12(n + 1, m + 1);
    end
end

%Estimate the correlation coefficient of RV1 and RV2
rho12 = (mu12 - (mu1 * mu2)) / (sig1 * sig2);
```

```
%Compute the joint PMF of rv1 and rv3
p13 = zeros(101, 101);
for n = 1:N
    i = RV3(n) + 1;
    j = RV1(n) + 1;
    p13(i,j) = p13(i,j) + 1;
end
p13 = p13 / N;

%Estimate the expected value of rv1 rv3
mu13 = 0;
for n = 0:100
    for m = 0:100
        mu13 = mu13 + n *m * p13(n + 1, m + 1);
    end
end

%Estimate the correlation coeffcient of RV1 and RV3
rho13 = (mu13 - (mu1 * mu3)) / (sig1 * sig3);
```

Figure 11: MATLAB code used to determine the above results for Question 2 part II

Question 3

Part I

To estimate the probability that **x** takes on a value between 20 and 70, the probability masses of the joint PMF of (**x,y**) were summed across all values of **y** where **x** is greater than or equal to 20, and less than or equal to 70. It is important to note that it was assumed that due to the given joint PMF being a 101 x 101 matrix, each random variable could take on integer values between 0 and 100. Also, to account for the fact that MATLAB indices start at 1, the sum was calculated over indices 21 to 71. The probability that **x** takes on a value between 20 and 70 is found below.

$$P[20 \leq \boldsymbol{x} \leq 70] = 0.9505$$

The MATLAB script used to compute this is displayed in Figure 12 below.

```
% ELEC 326
% Group Simulation Activity 2
% Question 3 - Part I
% Emma Chan, Charlotte Lombard, Jack Mason, Jake Moffat

% Prompt: Write a MATLAB code to estimate the probability that x takes
% values between 20 and 70.

% using H.mat, create a matrix with the same name, H
% RV1 = load('RV1.mat').RV1; sample load from Activity 1
H = load('H.mat').H;

% Estimate the probability that x takes values between 20 and 70
% Note: use indexes 21 and 71 as MatLab starts at 1, not 0
P_20_x_70 = sum(sum(H(21:71, :)));
```

Figure 12: MATLAB script for generating the joint PMF between x= [20 70]

Part II

To generate the conditional PMF of **x** with respect to certain **y** values (**y**=10 and **y**=40), two steps are needed. First, the marginal probability mass function of **y** must be calculated by summing the joint probability mass over the set of all its possible values of **x** in the support of **x** (the set of all possible values denoted by R_x).

$$p_y(y) = \sum_{xeR_x} p_{xy}(x, y)$$

Next, the conditional PDF can be calculated using the joint PMF of **x** and **y** when y meets the condition divided by the marginal PMF of **y**. Shown below:

$$p_{x|y}(x|y) = \frac{p_{xy}(x, y)}{p_y(y)}$$

The MATLAB code for the above calculations can be found in Figure 13 below. Please refer to the comments in code for further clarification on method or strategy.

```matlab
% ELEC 326
% Group Simulation Activity 2
% Question 3 - Part II
% Emma Chan, Charlotte Lombard, Jack Mason, Jake Moffat

% Prompt: Write a MATLAB code to plot the conditional PMF of p_x|y(x|40)
% and compare it with p_x|y(x|10)

% using H.mat, create a matrix with the same name, H - same as Part I
% RV1 = load('RV1.mat').RV1; sample load from Activity 1
H = load('H.mat').H;

% initialize p_x|y(x|40) and p_x|y(x|10)
Pxy_x40 = H(:, 41) / sum(H(:, 41)); % index 41 = 40
Pxy_x10 = H(:, 11) / sum(H(:, 11)); % index 11 = 10

% ---------------------------------------------------------------------
% Plot 1
% Plot the conditional PMF of p_x|y(x|40)
plot1 = figure('Name', 'Conditional PMF x|40');

%stem(x, y, 'Color', 'r', 'LineWidth', 2);
%[x] = meshgrid(0:100);
x = (0:100);
stem(x, Pxy_x40, 'Color', 'r', 'LineWidth', 1);
title('Conditional PMF given y = 40', 'FontWeight', 'normal');
set(gca, 'FontSize', 11);
axis1 = gca;
axis1.XLabel.String = 'x';
axis1.YLabel.String = 'p_{x|y}(x|40)';

% ---------------------------------------------------------------------
% Plot 2
% Plot the conditional PMF of p_x|y(x|10)
plot1 = figure('Name', 'Conditional PMF x|10');

stem(x, Pxy_x10, 'Color', 'b', 'LineWidth', 1);
title('Conditional PMF given y = 10', 'FontWeight', 'normal');
set(gca, 'FontSize', 11);
axis1 = gca;
axis1.XLabel.String = 'x';
axis1.YLabel.String = 'p_{x|y}(x|10)';
```

Figure 13: MATLAB script used to discover the conditional PMF and plot the results

The plots of the conditional PMFs for **y**=10 and **y**=40 are shown in figure 14 and figure 15 respectively. The first plot when **y**=10 has a distribution centered around **x**=30 while the second plot has a distribution centered around **x**=60. Plot 1 shows a distribution similar to a normal distribution but with multiple outliers and exceptions while the second plot illustrates a smoother distribution much closer to an ideal normal distribution.

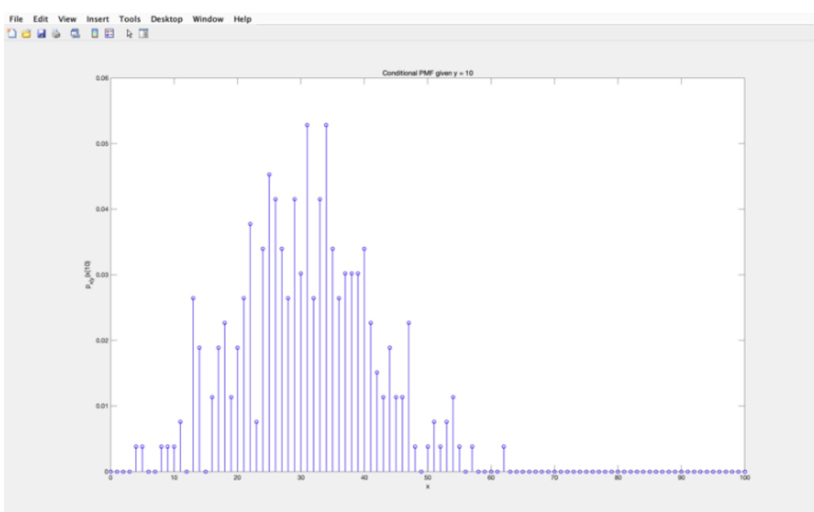

Figure 14: Plot showing the conditional PMF when y=10

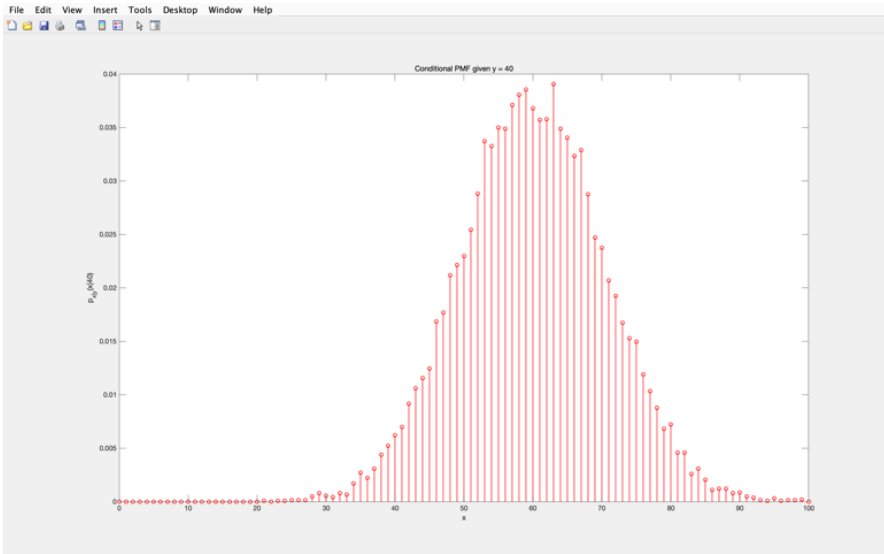

Figure 15: Plot showing the conditional PMF of y=40