% Euclidean Distance Calculation with Circle Visualization

% Define points

P1 = [1, 2]; % First point (x1, y1)

P2 = [4, 6]; % Second point (x2, y2)

% Calculate Euclidean Distance

euclideanDistance = sqrt(sum((p2 – p1).^2));

% Display the result

Fprintf(‘Euclidean Distance between points [%d, %d] and [%d, %d] is: %.2f\n’, …

P1(1), p1(2), p2(1), p2(2), euclideanDistance);

% Visualization

Figure;

Hold on;

% Plot points

Scatter(p1(1), p1(2), 100, ‘r’, ‘filled’); % Point 1

Scatter(p2(1), p2(2), 100, ‘b’, ‘filled’); % Point 2

% Add labels to points

Text(p1(1), p1(2), ‘ P1’, ‘VerticalAlignment’, ‘bottom’);

Text(p2(1), p2(2), ‘ P2’, ‘VerticalAlignment’, ‘bottom’);

% Draw straight line between points

Plot([p1(1), p2(1)], [p1(2), p2(2)], ‘k-‘, ‘LineWidth’, 1.5); % Straight line

% Plot circle

Theta = linspace(0, 2\*pi, 100); % Angle values for circle

X\_circle = p1(1) + euclideanDistance \* cos(theta); % X-coordinates of circle

Y\_circle = p1(2) + euclideanDistance \* sin(theta); % Y-coordinates of circle

Plot(x\_circle, y\_circle, ‘g—‘, ‘LineWidth’, 1.5); % Circle around p1

% Titles and grid

Title(‘Euclidean Distance with Circle Visualization’);

Grid on;

Axis equal; % Equal scaling for both axes

Hold off;