```
figure('NumberTitle','off');
cieplot();
title('Question 2: Laboratory Regular Users', 'FontSize', 13);              %% -- CHANGE HERE
xlabel('X Value');
ylabel('Y Value');
hold on;

scatter(x_pre_expectedColors(1), y_pre_expectedColors(1), 60, 'black', 'Filled');↙
% draw GIVEN COLOR
scatter(x_pre_expectedColors(2), y_pre_expectedColors(2), 60, 'black');↙
% draw EXPECTED COLOR
scatter(x_pre_expectedColors(3), y_pre_expectedColors(3), 60, 'black');↙
% draw EXPECTED COLOR

% Draw every pair of responses.
for i = 1 : height(laboratoryResults)
    %% Check if any Component is white or empty Answer
    sColor = str2num(cell2mat(laboratoryResults.second_color(i)));
    tColor = str2num(cell2mat(laboratoryResults.third_color(i)));

    if sColor == 0 || tColor == 0
        whiteAnswers = [whiteAnswers ; laboratoryResults(i,:)];
        rowsToEliminate = [rowsToEliminate i];
        continue
    end

    %% First Color - C1
    sColor = str2num(cell2mat(laboratoryResults.second_color(i)));      % cell2mat serve para↙
converter cell para matriz de uint
    valuesSColor = [valuesSColor sColor];                               % store integers↙
values of answers
    sColor = sColor/360;                                                % scale value to↙
[0,1] instead of [0, 360]
    sColor = [sColor 1 1];                                             % form triplet↙
(sColor, 1, 1)
    hsv_C1 = sColor;
    sColor = hsv2rgb(sColor);                                           % hsv -> rgb, values↙
between [0,1]
    rgb_C1 = round([sColor(1)*255 sColor(2)*255 sColor(3)*255]);
    sColor = rgb2xyz(sColor);                                           % rgb -> xyz
    x_aux = sColor(1)/(sColor(1) + sColor(2) + sColor(3));              % x = X / (X + Y + Z)
    y_aux = sColor(2)/(sColor(1) + sColor(2) + sColor(3));              % x = Y / (X + Y + Z)
    x_values = [x_values x_aux];                                        % store coordinates X↙
and Y
    y_values = [y_values y_aux];

    %% Second Color - C2
    tColor = str2num(cell2mat(laboratoryResults.third_color(i)));
    valuesTColor = [valuesTColor tColor];                               % store integers↙
values of answers
    tColor = tColor/360;                                                % scale value to↙
[0,1] instead of [0, 360]
    tColor = [tColor 1 1];                                             % form triplet↙
(tColor, 1, 1)
    hsv_C2 = tColor;
    tColor = hsv2rgb(tColor);                                           % hsv -> rgb, values↙
between [0,1]
    rgb_C2 = round([tColor(1)*255 tColor(2)*255 tColor(3)*255]);
    tColor = rgb2xyz(tColor);                                           % rgb -> xyz
    x_aux = tColor(1)/(tColor(1) + tColor(2) + tColor(3));              % x = X / (X + Y + Z)
    y_aux = tColor(2)/(tColor(1) + tColor(2) + tColor(3));              % x = Y / (X + Y + Z)
    x_values = [x_values x_aux];                                        % store coordinates X↙
and Y
```

```
    y_values = [y_values y_aux];

    %% Categorize Colors - Bins
    for j = 1 : length(colorBins)
        valuesofx = colorBins(j).XData;
        valuesofy = colorBins(j).YData;
        for a = 1 : length(valuesofx)
            if foundC1 == 0 && ((x_values(1) > (valuesofx(a) - 0.03) && x_values(1) <↙
(valuesofx(a) + 0.03)) && (y_values(1) > (valuesofy(a) - 0.03) && y_values(1) < (valuesofy(a)↙
+ 0.03)))
                color_C1 = colorBins(j).Tag;
                foundC1 = 1;
            end
            if foundC2 == 0 && ((x_values(2) > (valuesofx(a) - 0.03) && x_values(2) <↙
(valuesofx(a) + 0.03)) && (y_values(2) > (valuesofy(a) - 0.03) && y_values(2) < (valuesofy(a)↙
+ 0.03)))
                color_C2 = colorBins(j).Tag;
                foundC2 = 1;
            end
        end
    end

    if foundC1 == 0 && foundC2 == 0
        C1_name = [C1_name ; 'NA       '];
        C2_name = [C2_name ; 'NA       '];
    else if foundC1 == 1 && foundC2 == 0
            C1_name = [C1_name ; color_C1];
            C2_name = [C2_name ; 'NA       '];
        else if foundC1 == 0 && foundC2 == 1
                C1_name = [C1_name ; 'NA       '];
                C2_name = [C2_name ; color_C2];
            else
                C1_name = [C1_name ; color_C1];
                C2_name = [C2_name ; color_C2];
            end
        end
    end
    foundC1 = 0; foundC2 = 0;

    %% Difference between C1/C2 and expected colors C1/C2
    diff_c1_expected = round(pdist([[x_values(1) y_values(1)] ; [x_pre_expectedColors(2)↙
y_pre_expectedColors(2)]]), 2);  % [given -C1- C2]
    diff_c2_expected = round(pdist([[x_values(2) y_values(2)] ; [x_pre_expectedColors(3)↙
y_pre_expectedColors(3)]]), 2);  % [given C1 -C2-]
    distance_expectedC1C2 = [distance_expectedC1C2 ; diff_c1_expected + diff_c2_expected];

    %% Difference between C1/C2 blended onto Color Models against pre-calc values
    %%%%%% Blend-it in HSV
    sColor_hsv = [(hsv_C1(1)*360) hsv_C1(2) hsv_C1(3)]; tColor_hsv = [(hsv_C2(1)*360) hsv_C2↙
(2) hsv_C2(3)];
    diff_angles = abs(sColor_hsv(1) - tColor_hsv(1));
    if diff_angles > 180
        angle_small = (360 - diff_angles);  % smallest angle
        angle_small_half = angle_small / 2;
        sum_major = max([sColor_hsv(1) tColor_hsv(1)]) + angle_small_half;
        if sum_major > 360
            angle = rem((max([sColor_hsv(1) tColor_hsv(1)]) + angle_small_half), 360);
        else
            angle = max([sColor_hsv(1) tColor_hsv(1)]) + angle_small_half;
        end
    else
        angle = min([sColor_hsv(1) tColor_hsv(1)]) + (diff_angles / 2);
    end
```

```
    rColor = rgb2xyz(hsv2rgb([angle/360 sColor_hsv(2) sColor_hsv(3)]));↙
% hsv -> rgb -> xyz -> compare
    x_aux  = rColor(1)/(rColor(1) + rColor(2) + rColor(3)); y_aux = rColor(2)/(rColor(1) +↙
rColor(2) + rColor(3));
    distance_HSV = [distance_HSV; round(pdist([[x_aux y_aux]; [x_pre_models(1) y_pre_models↙
(1)]]),2)];
    x_values_HSV = [x_values_HSV x_aux]; y_values_HSV = [y_values_HSV y_aux];

    %%%%%% Blend-it in CIE-LCh (XYZ -> Lab -> LCh)
    sColor_lch = sColor; tColor_lch = tColor;
    sColor_lch = applycform(sColor_lch, cformXYZ_Lab); tColor_lch = applycform(tColor_lch,↙
cformLab_LCh);

    l_aux = (abs(sColor_lch(1) - tColor_lch(1)) / 2) + min([sColor_lch(1) tColor_lch(1)]);  %↙
diff between colors, and add half to the smallest
    c_aux = (abs(sColor_lch(2) - tColor_lch(2)) / 2) + min([sColor_lch(2) tColor_lch(2)]);
    diff_angles = abs(sColor_lch(3) - tColor_lch(3));
    if diff_angles > 180
        angle_small = (360 - diff_angles);  % smallest angle
        angle_small_half = angle_small / 2;
        sum_major = max([sColor_lch(3) tColor_lch(3)]) + angle_small_half;
        if sum_major > 360
            h_aux = rem((max([sColor_lch(3) tColor_lch(3)]) + angle_small_half), 360);
        else
            h_aux = max([sColor_lch(3) tColor_lch(3)]) + angle_small_half;
        end
    else
        h_aux = min([sColor_lch(3) tColor_lch(3)]) + (diff_angles / 2);
    end
    rColor = applycform(applycform([l_aux c_aux h_aux], cformLCh_Lab), cformLab_XYZ);
    x_aux  = rColor(1)/(rColor(1) + rColor(2) + rColor(3)); y_aux = rColor(2)/(rColor(1) +↙
rColor(2) + rColor(3));
    distance_LCh = [distance_LCh; round(pdist([[x_aux y_aux]; [x_pre_models(2) y_pre_models↙
(2)]]), 2)];
    x_values_LCh = [x_values_LCh x_aux]; y_values_LCh = [y_values_LCh y_aux];

    %%%%%% Blend-it in CMYK
    sColor_cmyk = sColor; tColor_cmyk = tColor;
    sColor_cmyk = applycform(applycform(sColor_cmyk, cformXYZ_RGB), cformRGB_CMYK);↙
tColor_cmyk = applycform(applycform(tColor_cmyk, cformXYZ_RGB), cformRGB_CMYK);
    sColor_cmyk = [sColor_cmyk(1) sColor_cmyk(2) sColor_cmyk(3)]; tColor_cmyk = [tColor_cmyk↙
(1) tColor_cmyk(2) tColor_cmyk(3)]; %Exclude 'K' component
    c_aux = (abs(sColor_cmyk(1) - tColor_cmyk(1)) / 2) + min([sColor_cmyk(1) tColor_cmyk↙
(1)]);
    m_aux = (abs(sColor_cmyk(2) - tColor_cmyk(2)) / 2) + min([sColor_cmyk(2) tColor_cmyk↙
(2)]);
    y_aux = (abs(sColor_cmyk(3) - tColor_cmyk(3)) / 2) + min([sColor_cmyk(3) tColor_cmyk↙
(3)]);
    rColor = applycform(applycform([c_aux m_aux y_aux 0], cformCMYK_RGB), cformRGB_XYZ);
    x_aux  = rColor(1)/(rColor(1) + rColor(2) + rColor(3)); y_aux = rColor(2)/(rColor(1) +↙
rColor(2) + rColor(3));
    distance_CMYK = [distance_CMYK; round(pdist([[x_aux y_aux]; [x_pre_models(3) y_pre_models↙
(3)]]), 2)];
    x_values_CMYK = [x_values_CMYK x_aux]; y_values_CMYK = [y_values_CMYK y_aux];

    %%%%%% Blend-it in RGB
    sColor_rgb = sColor; tColor_rgb = tColor;
    sColor_rgb = applycform(sColor_rgb, cformXYZ_RGB); tColor_rgb = applycform(tColor_rgb,↙
cformXYZ_RGB);
    r_aux = (abs(sColor_rgb(1) - tColor_rgb(1)) / 2) + min([sColor_rgb(1) tColor_rgb(1)]);
    g_aux = (abs(sColor_rgb(2) - tColor_rgb(2)) / 2) + min([sColor_rgb(2) tColor_rgb(2)]);
    b_aux = (abs(sColor_rgb(3) - tColor_rgb(3)) / 2) + min([sColor_rgb(3) tColor_rgb(3)]);
    rColor = applycform([r_aux g_aux b_aux], cformRGB_XYZ);
```

```
    x_aux  = rColor(1)/(rColor(1) + rColor(2) + rColor(3)); y_aux = rColor(2)/(rColor(1) +↙
rColor(2) + rColor(3));
    distance_RGB = [distance_RGB; round(pdist([[x_aux y_aux]; [x_pre_models(4) y_pre_models↙
(4)]]), 2)];
    x_values_RGB = [x_values_RGB x_aux]; y_values_RGB = [y_values_RGB y_aux];

    %%%%%% Blend-it in CIE-Lab
    sColor_lab = sColor; tColor_lab = tColor;
    sColor_lab = applycform(sColor_lab, cformXYZ_Lab); tColor_lab = applycform(tColor_lab,↙
cformXYZ_Lab);
    l_aux = (abs(sColor_lab(1) - tColor_lab(1)) / 2) + min([sColor_lab(1) tColor_lab(1)]);
    a_aux = (abs(sColor_lab(2) - tColor_lab(2)) / 2) + min([sColor_lab(2) tColor_lab(2)]);
    b_aux = (abs(sColor_lab(3) - tColor_lab(3)) / 2) + min([sColor_lab(3) tColor_lab(3)]);
    rColor = applycform([l_aux a_aux b_aux], cformLab_XYZ);
    x_aux  = rColor(1)/(rColor(1) + rColor(2) + rColor(3)); y_aux = rColor(2)/(rColor(1) +↙
rColor(2) + rColor(3));
    distance_Lab = [distance_Lab; round(pdist([[x_aux y_aux]; [x_pre_models(5) y_pre_models↙
(5)]]), 2)];
    x_values_Lab = [x_values_Lab x_aux]; y_values_Lab = [y_values_Lab y_aux];

    scatter(x_values, y_values, 50, 'white');                          %draw two responses
    plot(x_values, y_values, 'Color', 'black');                       %draw relations between↙
answers
    x_values = [];                                                 %clean the arrays
    y_values = [];
end
hold off;

saveas(gcf, fullfile(path, 'lab_regularUsers'), 'png'); close;
laboratoryResults(rowsToEliminate, :) = []; rowsToEliminate = [];

% Centroids of Color Models
centroid_HSV    = [centroid_HSV  ; [round(mean(x_values_HSV),2) round(mean(y_values_HSV),↙
2)]];
centroid_LCh    = [centroid_LCh  ; [round(mean(x_values_LCh),2) round(mean(y_values_LCh),↙
2)]];
centroid_CMYK   = [centroid_CMYK ; [round(mean(x_values_CMYK),2) round(mean(y_values_CMYK),↙
2)]];
centroid_RGB    = [centroid_RGB  ; [round(mean(x_values_RGB),2) round(mean(y_values_RGB),↙
2)]];
centroid_Lab    = [centroid_Lab  ; [round(mean(x_values_Lab),2) round(mean(y_values_Lab),↙
2)]];

distance_centroid_HSV = [distance_centroid_HSV ; round(pdist([mean(x_values_HSV) mean↙
(y_values_HSV); [x_pre_models(1) y_pre_models(1)]]), 2)];
distance_centroid_LCh = [distance_centroid_LCh ; round(pdist([mean(x_values_LCh) mean↙
(y_values_LCh); [x_pre_models(2) y_pre_models(2)]]), 2)];
distance_centroid_CMYK = [distance_centroid_CMYK ; round(pdist([mean(x_values_CMYK) mean↙
(y_values_CMYK); [x_pre_models(3) y_pre_models(3)]]), 2)];
distance_centroid_RGB = [distance_centroid_RGB ; round(pdist([mean(x_values_RGB) mean↙
(y_values_RGB); [x_pre_models(4) y_pre_models(4)]]), 2)];
distance_centroid_Lab = [distance_centroid_Lab ; round(pdist([mean(x_values_Lab) mean↙
(y_values_Lab); [x_pre_models(5) y_pre_models(5)]]), 2)];

% Catenate all Tables                -- CHANGE HERE
diffs_table = table(distance_expectedC1C2, distance_HSV, distance_LCh, distance_CMYK,↙
distance_RGB, distance_Lab);

if size(C1_name, 1) == 1
    colors_names = cell(1,2);
    colors_names(1,1) = cellstr(C1_name);
    colors_names(1,2) = cellstr(C2_name);
    colors_names = cell2table(colors_names);
```

```
else
    colors_names = table(C1_name, C2_name);
end

C1_name = []; C2_name = [];

laboratoryResults = [laboratoryResults colors_names diffs_table];

% Plot Results for each Color Model
figure('NumberTitle','off');
cieplot();
title('Question 2: Laboratory Responses According to HSV Model', 'FontSize', 13);↙
%% -- CHANGE HERE
xlabel('X Value');
ylabel('Y Value');
hold on;
scatter(x_pre_models(1), y_pre_models(1), 50, 'black', 'Filled');          % Draw expected↙
response for this model
scatter(x_values_HSV, y_values_HSV, 50, 'white');                          % Draw responses↙
mixed in HSV
hold off;
saveas(gcf, fullfile(path, 'lab_HSVresponses'), 'png'); close;

figure('NumberTitle','off');
cieplot();
title('Question 2: Laboratory Responses According to CIE-LCh Model', 'FontSize', 13);↙
%% -- CHANGE HERE
xlabel('X Value');
ylabel('Y Value');
hold on;
scatter(x_pre_models(2), y_pre_models(2), 50, 'black', 'Filled');          % Draw expected↙
response for this model
scatter(x_values_LCh, y_values_LCh, 50, 'white');                          % Draw responses↙
mixed in LCh
hold off;
saveas(gcf, fullfile(path, 'lab_LChresponses'), 'png'); close;

figure('NumberTitle','off');
cieplot();
title('Question 2: Laboratory Responses According to CMYK Model', 'FontSize', 13);↙
%% -- CHANGE HERE
xlabel('X Value');
ylabel('Y Value');
hold on;
scatter(x_pre_models(3), y_pre_models(3), 50, 'black', 'Filled');          % Draw expected↙
response for this model
scatter(x_values_CMYK, y_values_CMYK, 50, 'white');                          % Draw responses↙
mixed in CMYK
hold off;
saveas(gcf, fullfile(path, 'lab_CMYKresponses'), 'png'); close;

figure('NumberTitle','off');
cieplot();
title('Question 2: Laboratory Responses According to RGB Model', 'FontSize', 13);↙
%% -- CHANGE HERE
xlabel('X Value');
ylabel('Y Value');
hold on;
scatter(x_pre_models(4), y_pre_models(4), 50, 'black', 'Filled');          % Draw expected↙
response for this model
scatter(x_values_RGB, y_values_RGB, 50, 'white');                          % Draw responses↙
mixed in RGB
hold off;
```

```
saveas(gcf, fullfile(path, 'lab_RGBresponses'), 'png'); close;

figure('NumberTitle','off');
cieplot();
title('Question 2: Laboratory Responses According to CIE-Lab Model', 'FontSize', 13);↙
%% -- CHANGE HERE
xlabel('X Value');
ylabel('Y Value');
hold on;
scatter(x_pre_models(5), y_pre_models(5), 50, 'black', 'Filled');          % Draw expected↙
response for this model
scatter(x_values_Lab, y_values_Lab, 50, 'white');                          % Draw responses↙
mixed in Lab
hold off;
saveas(gcf, fullfile(path, 'lab_Labresponses'), 'png'); close;

% Clean all the tables!
x_values_HSV = []; y_values_HSV = []; x_values_LCh = []; y_values_LCh = []; x_values_CMYK =↙
[]; y_values_CMYK = []; x_values_RGB = []; y_values_RGB = []; x_values_Lab = []; y_values_Lab↙
= [];
distance_HSV = []; distance_LCh = []; distance_CMYK = []; distance_RGB = []; distance_Lab =↙
[];
distance_expectedC1C2 = [];
```