EECE 5612 Detectathon
Stav Rones
4.13.2022

For door 1, the 3 cases were analyzed by finding the MSE between the observed average arrival rate over the given time period and the expected rate.

```
Door 1:
   MSE a: (1.28-1)^2 = 0.08
   MSE b: (1.28-2)^2 = 0.52
   MSE c: = (1.00-1)^2 + (2.03-2)^2 = 0.00076
   MSE is minimized for case c (break at t0_i=7378)
```

Case a compared the rate from 0 to Tn to the working rate = 1. Case b compared the rate from 0 to Tn to the malfunctioning rate = 2. Case 3 was the sum of the MSE between the working rate from 0 to t0 and the malfunctioning rate from t0 to tn. Since the MSE was minimized for case 3, this was determined to be the correct scenario.

To find the best t0, MMSE was used again where t0 was iterated from 0 to tn and the t0 that was chosen minimized the sum of the MSE of the observed rates before and after t0 compared to the known rates.

```
t0_ML = 0;
e_min = inf;
for i = 2:n

    lambda1_i = 1/mean(y1(1:i-1));
    lambda2_i = 1/mean(y1(i:n));

    e = (1 - lambda1_i)^2 + (2 - lambda2_i)^2;

    if (e < e_min)
        e_min = e;
        lambda1_ML = lambda1_i;
        lambda2_ML = lambda2_i;
        t0_ML = i;
    end

end
```

The index of T0 was found to be 7290, which is close to the t0 in the previous part (7478). At this t0, the average rates of before and after t0 were 1 and 2 with a MSE of .000003.

```
t0_i known lambdas: 7290 with squared error = 0.000003
    lambda_1:   1.00
    lambda_2:   2.00
```

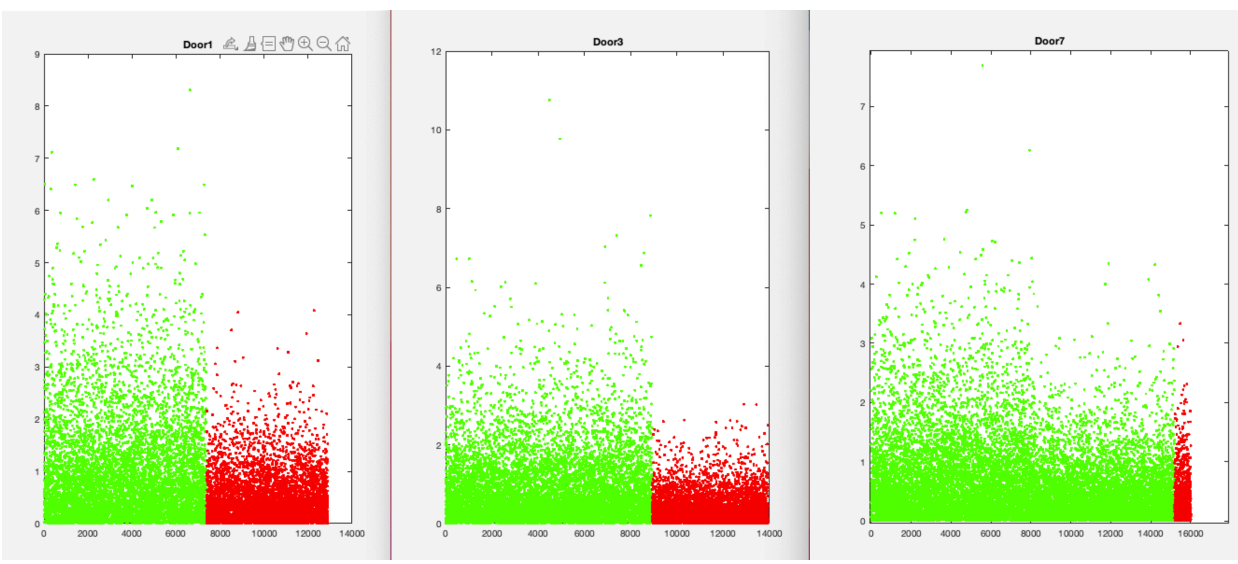
When the rates were unknown for the working and malfunctioning cases, t0 was iterated and chosen for the index that maximized the squared difference between the average rate before and after t0. This yielded poor results, however, so instead of looking at the entire dataset before and after t0, a window of radius 100 (201 window size) was used to compare the rates. This yielded much better results, and found a t0 at 7378.

```
t0_i unknown lambdas: 7378 with window size 201:
   lambda_1:   1.06
   lambda_2:   2.22
   squared difference = 1.341110
```

This same method was used to determine if the other doors had a malfunction. The following max squared difference for 100 points before and after t0 for each door is shown below:

```
Max diff for door 1: 1.34111
Max diff for door 2: 0.50873
Max diff for door 3: 1.94089
Max diff for door 4: 0.21691
Max diff for door 5: 0.45577
Max diff for door 6: 0.26993
Max diff for door 7: 1.24839
Max diff for door 8: 0.51945
Max diff for door 9: 0.41734
Max diff for door 10: 0.60242
Failures occur at doors 1, 3, and 7
```

The 3 biggest differences occur at doors 1, 3 and 7, as visualized below.



APPENDIX:

```
function detectathon

    clc;
    close all;

    % --------------------- Determine case 1, 2 or 3 ----------------------

    y1      = load("door1.mat").y;  % row vector of interarrival times
    n       = length(y1);           % 12,894
    t0      = (7*24*.73)*60;        % Possible minute of malfunciton

    % find t0 index
    t0_index = 0;
    tn = 0;

    for i = 1:n
        tn = tn + y1(i);
        if (tn >= t0)
            t0_index = i;
            break;
        end
    end

    lambda_n  = 1/mean(y1);
    lambda_0_t0 = 1/mean(y1(1:t0_index-1));
    lambda_t0_n = 1/mean(y1(t0_index:n));

    MSE_a = (lambda_n - 1)^2;
    MSE_b = (lambda_n - 2)^2;
    MSE_c = (lambda_0_t0 - 1)^2 + (lambda_t0_n - 2)^2;

    fprintf("Door 1:\n");
    fprintf("   MSE a: (%3.2f-%i)^2 = %3.2f\n", ...
        lambda_n, 1, MSE_a);
    fprintf("   MSE b: (%3.2f-%i)^2 = %3.2f\n", ...
        lambda_n, 2, MSE_b);
    fprintf("   MSE c: = (%3.2f-1)^2 + (%3.2f-2)^2 = %6.5f\n", lambda_0_t0,
lambda_t0_n, MSE_c);
    fprintf("   MSE is minimized for case c (break at t0_i=%i)\n", t0_index)

    plot(1:n, y1', "g.")
    hold on;
    plot(t0_index:n, y1(t0_index:n)', "r.")
    title("Door 1")

    % ---------------------- find t0 known lambdas -----------------------

    t0_ML = 0;
    e_min = inf;
    for i = 2:n

        lambda1_i = 1/mean(y1(1:i-1));
        lambda2_i = 1/mean(y1(i:n));
```

```
        e = (1 - lambda1_i)^2 + (2 - lambda2_i)^2;

        if (e < e_min)
            e_min = e;
            lambda1_ML = lambda1_i;
            lambda2_ML = lambda2_i;
            t0_ML = i;
        end

    end

    fprintf("   t0_i known lambdas: %i with squared error = %6.6f\n", ...
        t0_ML, e_min);
    fprintf("       lambda_1: %6.2f\n       lambda_2: %6.2f\n", ...
        lambda1_ML, lambda2_ML);

    % ---------------------- find t0 unknown lambdas ----------------------

    t0_ML = 0;
    diff_max = 0;
    radius = 100;

    for i = radius+1:n-radius

        y_pre = y1(i-radius:i-1);
        y_post = y1(i:i+radius);

        m1 = 1/mean(y_pre);
        m2 = 1/mean(y_post);
        diff = (m2 - m1)^2;

        if (diff > diff_max)
            diff_max = diff;
            t0_ML = i;
            lambda1_ML = m1;
            lambda2_ML = m2;
        end

    end

    fprintf("   t0_i unknown lambdas: %i with window size %i:\n", t0_ML,
radius*2+1);
    fprintf("      lambda_1: %6.2f\n      lambda_2: %6.2f\n", ...
        lambda1_ML, lambda2_ML);
    fprintf("      squared difference = %6.6f\n", diff_max);

    % ---------------------- Which cameras break? ----------------------

    for j = 1:10

        filename = "door" + j + ".mat";
        y = load(filename).y;
        n = length(y);
```

```matlab
        t0_ML = 0;
        diff_max = 0;
        radius = 100;

        for i = radius+1:n-radius

            y_pre = y(i-radius:i-1);
            y_post = y(i:i+radius);

            m1 = 1/mean(y_pre);
            m2 = 1/mean(y_post);
            diff = (m2 - m1)^2;

            if (diff > diff_max)
                diff_max = diff;
                t0_ML = i;
                lambda1_ML = m1;
                lambda2_ML = m2;
            end

        end

        fprintf("Max diff for door %i: %6.5f\n", j, diff_max);

        if (diff_max > 1)
            plot(1:n, y', "g.")
            hold on;
            plot(t0_ML:n, y(t0_ML:n)', "r.")
            title("Door" + j)
            figure()
        end

    end

    fprintf("Failures occur at doors 1, 3, and 7\n\n");

end

function ret = poisson(k, lambda)
    ret = (lambda^k*exp(-lambda))/factorial(k);
end
```