```
clc; clear; close all;
```

```
% read data
data_1 = readmatrix("week1.csv");
data_2 = readmatrix("week2.csv");
data_ref = readmatrix("reference.xlsx");
```

```
% get ppg data
g_ppg_1 = data_1(:,1); ir_ppg_1 = data_1(:,2);
g_ppg_2 = data_2(:,1); ir_ppg_2 = data_2(:,2);
```

```
%process_ppg
[g_ppg_1_pro,ir_ppg_1_pro] = PPG_process(g_ppg_1,ir_ppg_1);
```

```
警告: 矩阵接近奇异值，或者缩放不良。结果可能不准确。RCOND =  2.100467e-16。
警告: 矩阵接近奇异值，或者缩放不良。结果可能不准确。RCOND =  2.100467e-16。
```

```
[g_ppg_2_pro,ir_ppg_2_pro] = PPG_process(g_ppg_2,ir_ppg_2);
```

```
警告: 矩阵接近奇异值，或者缩放不良。结果可能不准确。RCOND =  2.100467e-16。
警告: 矩阵接近奇异值，或者缩放不良。结果可能不准确。RCOND =  2.100467e-16。
```

```

% calculate PPT
PTT_1 = Cal_PTT(g_ppg_1_pro,ir_ppg_1_pro);
PTT_2 = Cal_PTT(g_ppg_2_pro,ir_ppg_2_pro);
avg_PTT_1 = Cal_avgPTT(PTT_1, data_ref(3, 2:8));
avg_PTT_2 = Cal_avgPTT(PTT_2, data_ref(8, 2:10));

% extract BP
SBP1 = data_ref(1, 2:8); SBP2 = data_ref(6,2:10);
DBP1 = data_ref(2,2:8);  DBP2 = data_ref(7,2:10);

% fit and predict
[pre_SBP_1, pre_DBP_1] = fit_and_predict(avg_PTT_1, SBP1, DBP1, avg_PTT_2);
[pre_SBP_2, pre_DBP_2] = fit_and_predict(avg_PTT_2, SBP2, DBP2, avg_PTT_1);

% calculate MAE
[mae1_SBP, mae1_DBP] = Cal_MAE(pre_SBP_1, SBP2, pre_DBP_1, DBP2);
[mae2_SBP, mae2_DBP] = Cal_MAE(pre_SBP_2, SBP1, pre_DBP_2, DBP1);

fprintf('MAE for SBP (Week 1 on Week 2): %.2f\n', mae1_SBP);
```

```
MAE for SBP (Week 1 on Week 2): 14.24
```

```
fprintf('MAE for DBP (Week 1 on Week 2): %.2f\n', mae1_DBP);
```

```
MAE for DBP (Week 1 on Week 2): 8.89
```

```
fprintf('MAE for SBP (Week 2 on Week 1): %.2f\n', mae2_SBP);
```

```
MAE for SBP (Week 2 on Week 1): 14.71
```

```
fprintf('MAE for DBP (Week 2 on Week 1): %.2f\n', mae2_DBP);
```

```
MAE for DBP (Week 2 on Week 1): 9.68
```

```
function [g_ppg_pro, ir_ppg_pro] = PPG_process(g_ppg_pro, ir_ppg_pro)
 % DC normalization
 g_ppg_pro = g_ppg_pro ./ repmat(mean(g_ppg_pro, 2) + 1e-6, [1, size(g_ppg_pro,
2)]) - 1;
 ir_ppg_pro = ir_ppg_pro ./ repmat(mean(ir_ppg_pro, 2) + 1e-6, [1, size(ir_ppg_pro,
2)]) - 1;

 % 滤波 [0.5 4]
 fps = 60;
 [b, a] = butter(8, [0.5, 4] / (fps / 2), 'bandpass');
 g_ppg_pro = filtfilt(b, a, g_ppg_pro);
 ir_ppg_pro = filtfilt(b, a, ir_ppg_pro);
end

function PTT = Cal_PTT(g_ppg, ir_ppg)
 g_ppg = resample(g_ppg, 120, 60);
 ir_ppg = resample(ir_ppg, 120, 60);
 [~, g_loc] = findpeaks(g_ppg);
 [~, ir_loc] = findpeaks(ir_ppg);

 PTT = [];
 for i = 1:length(g_loc)
 [~, idx] = min(abs(g_loc(i) - ir_loc));
 if ir_loc(idx) < g_loc(i) && ir_loc(idx) > g_loc(i) - 10
 PTT = [PTT; g_loc(i) - ir_loc(idx)];
 end
 end
 PTT(PTT == 0) = []; %去除为 0 的部分
end

function avgPTT = Cal_avgPTT(PTT, intervals)
 start_idx = 1;
 avgPTT = zeros(1, length(intervals));
 for i = 1:length(intervals)
    end_idx = start_idx + intervals(i) - 1;
 if end_idx > length(PTT)
 end_idx = length(PTT);
 end
 if ~isnan(mean(PTT(start_idx:end_idx)))
 avgPTT(i) = mean(PTT(start_idx:end_idx));
 end
 start_idx = end_idx + 1;
 end
end
function [predict_SBP, predict_DBP] = fit_and_predict(avgPTT, SBP, DBP, avgPTT_new)
 coefficients_SBP = polyfit(avgPTT, SBP, 1);
 coefficients_DBP = polyfit(avgPTT, DBP, 1);
 predict_SBP = polyval(coefficients_SBP, avgPTT_new);
 predict_DBP = polyval(coefficients_DBP, avgPTT_new);
end

function [mae_SBP, mae_DBP] = Cal_MAE(predict_SBP, actual_SBP, predict_DBP,
actual_DBP)
 mae_SBP = mean(abs(predict_SBP - actual_SBP));
 mae_DBP = mean(abs(predict_DBP - actual_DBP));
end
```