```
 % Solution Exercise 1 CMCN2022/23
 close all
 clear
```

a. Use function readtable() to read the data from the file into a Matlab table.

```
 fileName = '..\Data\GC_MS_Data_exercise_1.txt'; % define file name
 MSDataTable = readtable(fileName,...
     'Delimiter','\t',... % define delimiter
     'ReadVariableNames',true,... % file has a header line
     'ReadRowNames', false... % don't use first column as row names
     );
```

b. Create a numeric (double) matrix from the metabolite abundance data of the table.

```
 idx=find(contains(MSDataTable.Properties.VariableNames, 'INTERNALSTAND',
'IgnoreCase',true));
 AbundanceMatrix = table2array(MSDataTable(:,idx:end));
 % alternative way:
 % AbundanceMatrix = MSDataTable{:,7:end};
```

c. Use function histogram() to plot a histogram summarizing the distribution of peak height across all metabolites under control conditions at time point 1 with 30 bins. Next, overlay the histogram created before with a histogram showing the distribution of peak height across all metabolites under cold stress at time point 1 using the same bins as before.

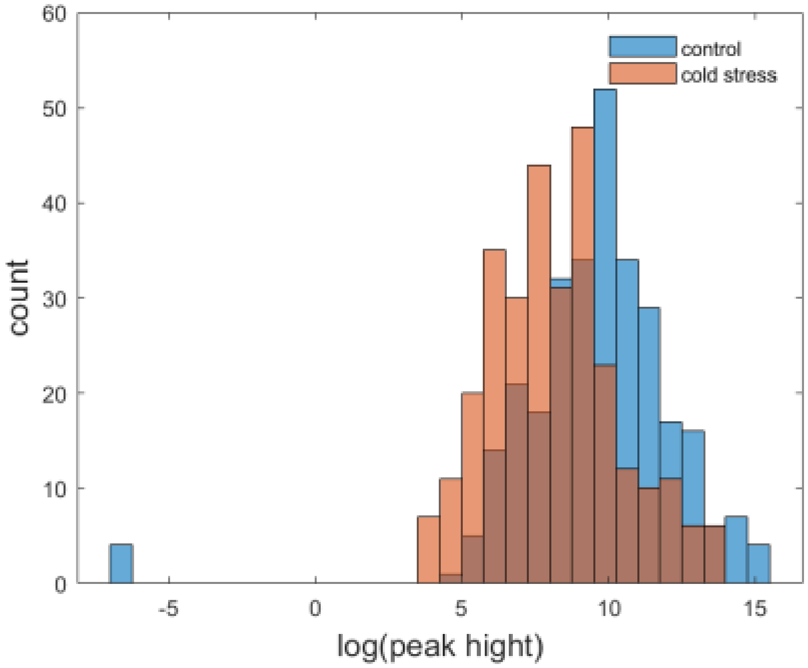
```
 control_tp1_idx=intersect(... % idx where both aguments are true
     find(strcmp(MSDataTable.stress,'control')),... % condition is control
     find(MSDataTable.timePoint==9)); % time point is 1

 figure
 h1=histogram(log(AbundanceMatrix(control_tp1_idx,:)),... % rows where condition
is control and tp is 1 found before, ':' means all, we use all
columns/metabolites
     'NumBins',30); % we want to use 30 bins
 xlabel('log(peak hight)','FontSize',14)
 ylabel('count','FontSize',14)
 hold on

 cold_tp1_idx=intersect(... % idx where both aguments are true
     find(strcmp(MSDataTable.stress,'cold stress')),... % condition is cold
stress
     find(MSDataTable.timePoint==9)); % time point is 1

 h2=histogram(log(AbundanceMatrix(cold_tp1_idx,:)),... % rows where condition is
cold stress and tp is 1 found before, all columns/metabolites
     'BinEdges',h1.BinEdges); % same bins as before
```

```
legend('control','cold stress')
legend boxoff
```



d. Calculate a new (double) matrix in which the replicates of each metabolite are pooled by calculating the mean across the 3 replicates, respectively.

```
% calculate the mean per metabolite and condition across three replicates
nTimePoints = 9; % number of time points
Conditions = unique(MSDataTable.stress); % list of conditions in the table
nConditions = length(unique(MSDataTable.stress)); % number of different
conditions

% initialize the new matrix
pooledAbundanceMatrix = zeros(0,... % we will add rows later
    size(AbundanceMatrix,2)); % number of metabolites is not changed

% for each condition and timepoint, calculate the mean per metabolite
for i=1:nConditions
    for t=1:nTimePoints

    % define the row indices where the three replicates are found
    rowIdx = intersect(... % idx where both aguments are true
    find(strcmp(MSDataTable.stress,Conditions(i))),... % condition
    find(MSDataTable.timePoint==t)); % time point

    pooledAbundanceMatrix(end+1,:) ... % append a row to the matrix
```

```
         = mean(AbundanceMatrix(rowIdx,:),1); % mean(...,1) means that we take
the mean for each column

     end
 end
```

e. Normalize the new matrix by dividing each value by the row sum.

```
% (sum(...,2) calculates the sum per row)
normalizedAbundanceMatrix = pooledAbundanceMatrix ./
sum(pooledAbundanceMatrix,2);
```

f. Create a new table from the normalized values, using the metabolite names as variable names. Add columns for time point and condition.

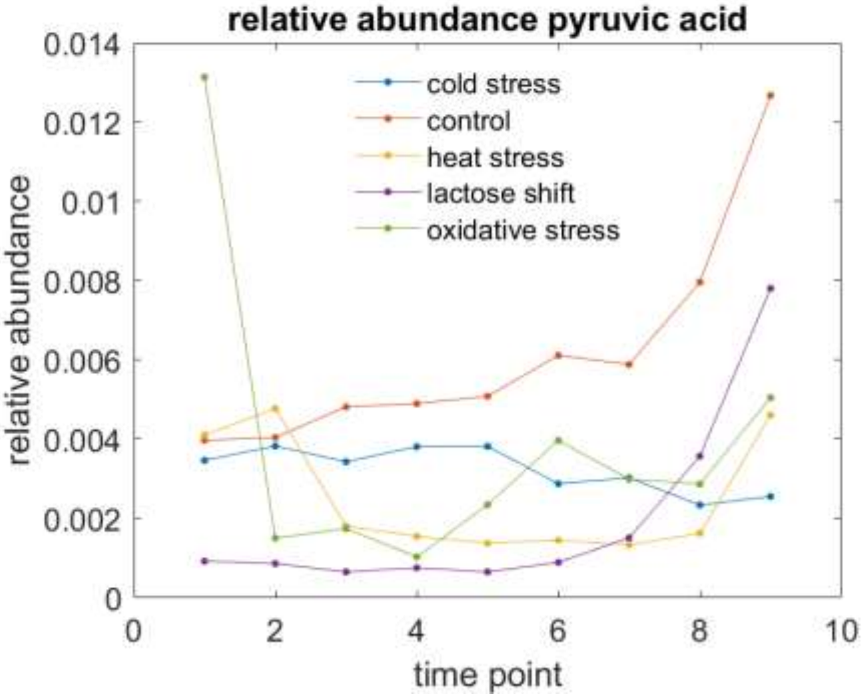
```
 normalizedAbundanceTable =
array2table(normalizedAbundanceMatrix,'VariableNames',MSDataTable.Properties.Var
iableNames(idx:end));
 ConditionsVectorReduced = repmat(Conditions',nTimePoints,1); % repeat condition
names to obtain matrix with condition names in columns, repeated in rows, nrows
equal to nTimePoints
 ConditionsVectorReduced = ConditionsVectorReduced(:); % vectorize
 tpVectorReduced = repmat((1:nTimePoints)',nConditions,1); % repeat time points
for each condition

 normalizedAbundanceTable =
[cell2table(ConditionsVectorReduced,'VariableNames',{'stressCondition'}),... %
add condition column
     array2table(tpVectorReduced,'VariableNames',{'timePoint'}),... % add time
point column
     normalizedAbundanceTable];
```

g. Create a new figure using function plot() showing the change in normalized abundance of Pyruvic acid over time for the different conditions. Do not forget to add a legend.

```
 figure
 for i=1:nConditions
     plot(1:nTimePoints,... % x axis values
      normalizedAbundanceTable.PyruvicAcid(strcmp(normalizedAbundanceTable.stres
sCondition,Conditions(i))),... % y-axis values
         '.-','MarkerSize',10) % use points combined by lines and increase size
of them
     hold on
 end
 set(gca,'FontSize',14)
 title('relative abundance pyruvic acid')
 xlabel('time point')
 ylabel('relative abundance')
```

```
legend(Conditions,'Location','north')
legend boxoff
```


relative abundance pyruvic acid
relative abundance
time point
cold stress
control
heat stress
lactose shift
oxidative stress