
```

function data=readExpData(file)
% 包寒吴霜 - 201728012503021 - Date: 20180413 (revised: 20180428)
% 设计思路: 每次读取一行, 若读到greenbox1或redbox1, 则初始化一个新的结构体变量trial;
% 若读到其他需要提取的信息, 则对trial变量的相应属性赋值, 直到读到crossITI_8时trial结束
fid=fopen(file,'r'); % 原始数据文件指针
trialNum=0; % Trial的编号
afterScale=0; % 探测Event Type是否为Display Scale后面的第一个Response, 初始化为0
%% 数据读取
while ~feof(fid) % 持续读取直到文件末尾
    % 每次读取一行
    line=fgetl(fid);
    linedata=strsplit(line,'\t'); % 按制表符\t拆分字符串
    if length(linedata)<6 % 判断是否为正式数据 (即是否有多列)
        continue % 非正式数据则跳到下一个循环
    end
    event_type=linedata{3}; % 原始数据里的Event Type列
    code_text=linedata{4}; % 原始数据里的Code列
    current_time=str2double(linedata{5})/10000; % 原始数据里的Time列
    if afterScale & regexp(event_type,'Response')
        % 提取Display Scale后面的第一个Response的RT
        RT=str2double(linedata{6})/10000; % 原始数据里的TTime列, 本身即为RT
        % RT也可以用current_time-trial.Trial_OnsetTime-display_time_temp计算
        afterScale=0; % 恢复探测开关为0
    end
    % < === Trial判断开始 === >
    % trial为结构体变量 (struct)
    % greenbox1或redbox1为每个Trial开始的标记, 用正则表达式判断
    if regexp(code_text,'greenbox1|redbox1')
        % Trial的编号、类型与开始时间
        trialNum=trialNum+1;
        trial.Trial=trialNum;
        trial.Trial_Type=regexp(code_text,'greenbox|redbox','match');
        trial.Trial_OnsetTime=current_time;
        % 其余变量按顺序初始化为缺失值NaN
        trial.ISI_Onset=nan;
        trial.pain_DisplayOnset=nan; trial.pain_RT=nan; trial.pain_Rating=nan;
        trial.vibration_DisplayOnset=nan; trial.vibration_RT=nan; trial.vibration_Rating=nan;
        trial.difficulty_DisplayOnset=nan; trial.difficulty_RT=nan; trial.difficulty_Rating=nan;
        trial.difference_DisplayOnset=nan; trial.difference_RT=nan; trial.difference_Rating=nan;
        trial.ITI_Onset=nan;
        % 按要求判断条件、提取数据, 用正则表达式'\d+'匹配Code一列中的数字 (被试的评分)
        elseif regexp(code_text,'crossISI_4_8')
            trial.ISI_Onset=current_time-trial.Trial_OnsetTime;
        elseif regexp(code_text,'Display Scale')
            display_time_temp=current_time-trial.Trial_OnsetTime;
            afterScale=1; % 遇到Display Scale之后, 开始探测Response (打开开关)
        elseif regexp(code_text,'This pain')
            trial.pain_DisplayOnset=display_time_temp;
            trial.pain_RT=RT;
            trial.pain_Rating=str2double(regexp(code_text,'\d+','match'));
        elseif regexp(code_text,'This vibration')
            trial.vibration_DisplayOnset=display_time_temp;
            trial.vibration_RT=RT;
            trial.vibration_Rating=str2double(regexp(code_text,'\d+','match'));
        elseif regexp(code_text,'Difficulty')

```

```

        trial.difficulty_DisplayOnset=display_time_temp;
        trial.difficulty_RT=RT;
        trial.difficulty_Rating=str2double(regex(code_text, '\d+', 'match'));
elseif regexp(code_text, 'Difference')
    trial.difference_DisplayOnset=display_time_temp;
    trial.difference_RT=RT;
    trial.difference_Rating=str2double(regex(code_text, '\d+', 'match'));
% crossITI_8为每个Trial结束的标记
elseif regexp(code_text, 'crossITI_8')
    trial.ITI_Onset=current_time-trial.Trial_OnsetTime;
    % 合并trial至data
    if exist('data', 'var')==0
        data=struct2table(trial); % 如果是第1个trial, 则先赋值
    else
        data=[data;struct2table(trial)]; % 如果是第2个以上的trial, 则合并数据
    end
end
% < === Trial判断结束 === >
end
%% 数据输出
fclose('all'); % 关闭所有文件
data=sortrows(data, {'Trial_Type', 'Trial'}, {'ascend', 'ascend'}); % 数据排序
disp(data);
end

```