```
function T=PreGetData(databases,savefile)
%PREGETDATA Extract data from TUEG database as a function
%
% Support function. Only needed if PreData2020122213119.xlsx is needed to be rerun.
% Creates an excel spreadsheet from a local copy to the TUEG database v.1.1.0 and v.1.2.0
% Output: PreData2020122213119.xlsx
%
%   Authors:
%       Michael Caiola (Michael.Caiola@fda.hhs.gov)
%       Meijun Ye (Meijun.Ye@fda.hhs.gov)
%
%   Disclaimer: This software and documentation (the "Software") were
%   developed at the Food and Drug Administration (FDA) by employees of
%   the Federal Government in the course of their official duties.
%   Pursuant to Title 17, Section 105 of the United States Code,
%   this work is not subject to copyright protection and is in the
%   public domain. Permission is hereby granted, free of charge, to any
%   person obtaining a copy of the Software, to deal in the Software
%   without restriction, including without limitation the rights to
%   use, copy, modify, merge, publish, distribute, sublicense, or sell
%   copies of the Software or derivatives, and to permit persons to
%   whom the Software is furnished to do so. FDA assumes no
%   responsibility whatsoever for use by other parties of the Software,
%   its source code, documentation or compiled executables, and makes
%   no guarantees, expressed or implied, about its quality,
%   reliability, or any other characteristic. Further, use of this code
%   in no way implies endorsement by the FDA or confers any advantage
%   in regulatory decisions. Although this software can be
%   redistributed and/or modified freely, we ask that any derivative
%   works bear some notice that they are derived from it, and any
%   modified versions bear some notice that they have been modified.

wid='MATLAB:table:ModifiedAndSavedVarnames';
warning('off',wid);

if nargin<2
    savefile=[];
end
if nargin<1
    databases=["D:\tuh_eeg\v1.1.0\edf","D:\tuh_eeg\v1.2.0\edf"];
end
%Select database location
if isempty(databases)
    databases=uigetdir('','Select Database');
end

%Savefile name
if isempty(savefile)
    savefile="GetData"+string(datetime('now','Format','MMddyyyyHHmm'))+".xlsx";
end
if isempty(regexpi(savefile,'.xlsx$'))
```

```
    savefile=[savefile,'.xlsx'];
end

%Grab input parameters
opts = detectImportOptions('session_types_v02.xlsx');

%Setup Table
varTypes=↙
{'string','uint32','string','string','double','categorical','string','string','string','s↙
tring','categorical','datetime','categorical','categorical','categorical'};
%may need to rethink subject type
varNames={'Location','Subject','Session','Flag','Age','Sex','mTBI↙
Evidence','Method','Medication↙
(s)','Notes','Montage','Date','EEGType','EEGSubtype','LTM_Routine'};

T=table('Size',[0 15],'VariableTypes',varTypes,'VariableNames',varNames);

try
progressbar('Folders','Montages','Subjects','Session')
catch
warning("Progressbar not found and will not be displayed. To display progressbar go to:↙
https://www.mathworks.com/matlabcentral/fileexchange/6922-progressbar?s_tid=srchtitle")
end


%Cycle through directory
dnum=1;
for database=databases
    d0=dir(database);
    for ii=3:length(d0)
        mont=d0(ii).name(4:end);
        d=dir(fullfile(database,d0(ii).name));
        for i=3:length(d)
            %Load Extra Data
            opts.Sheet=str2double(d(i).name)+1;
            adddata=readtable('session_types_v02.xlsx',opts);
            d1=dir(fullfile(database,d0(ii).name,d(i).name));
            for j=3:length(d1)
                subject=uint32(str2double(d1(j).name));
                d2=dir(fullfile(database,d0(ii).name,d(i).name,d1(j).name));
                for k=3:length(d2)
                    session=string(d2(k).name(1:4));
                    date=datetime(d2(k).name(6:end),'InputFormat','yyyy_MM_dd');
                    ad1=lower(string(adddata{ismember(adddata.PatientId,subject) &↙
ismember(adddata.Session,session),3}));
                    ad2=lower(string(adddata{ismember(adddata.PatientId,subject) &↙
ismember(adddata.Session,session),4}));
                    ad3=lower(string(adddata{ismember(adddata.PatientId,subject) &↙
ismember(adddata.Session,session),5}));

                    if isempty(ad1)
```

```
                        ad1=missing;
                        ad2=missing;
                        ad3=missing;
                    else
                        ad1=ad1{1};
                        ad2=ad2{1};
                        ad3=ad3{1};
                    end

                    d3=dir(fullfile(database,d0(ii).name,d(i).name,d1(j).name,d2(k).↙
name));
                    for kk=3:length(d3)
                        if regexp(d3(kk).name,'.*txt$')==1
                            break
                        end
                        if kk==length(d3) %no textfile found
                            warning("Missing Text File: "+fullfile(database,d0(ii).name,d↙
(i).name,d1(j).name,d2(k).name))
                            kk=-1;
                        end
                    end
                    if kk>0
                        location=fullfile(database,d0(ii).name,d(i).name,d1(j).name,d2↙
(k).name,d3(kk).name);
                        [flag,age,sex,evidence,method,medication,notes]=readdata↙
(location);
                    else
                        location=missing;
                        flag=missing;
                        age=missing;
                        sex=missing;
                        evidence=missing;
                        method=missing;
                        medication=missing;
                        notes=missing;
                    end
                    newT={location,subject,session,flag,age,sex,evidence,method,↙
medication,notes,mont,date,ad1,ad2,ad3};
                    T=[T;newT];
                    try
                    progressbar([],[],[],(k-2)/(length(d2)-2));
                    catch
                    end
                end
                try
                progressbar([],[],((j-2)+(i-3)*100)/((length(d)-2)*100),[])
                catch
                end
            end
        end
        try
        progressbar([],(ii-2)/(length(d0)-2),[],[])
        catch
        end
    end
    try
    progressbar(dnum/length(databases),0,[],[])
    catch
    end
    dnum=dnum+1;



end
try
progressbar(1);
catch
end
    varTypes={'categorical','categorical','categorical'};
    varNames={'EEG Type','EEG Subtype','LTM/Routine'};
    %T1=table('Size',[0 3],'VariableTypes',varTypes,'VariableNames',varNames);
    T=sortrows(T,'Subject');
    %Write data to excel
    writetable(T,savefile)
end
function [flag,age,sex,evidence,method,medication,notes]=readdata(file)
fileID=fopen(file,'r');
A=string(fscanf(fileID,'%c'));
fclose(fileID);
%age
%year old
age=regexpi(A,'(\d{1,3}|[a-zA-Z-]*\s[a-zA-Z-]+)(?=[-\s]*y(ear|o|r|\.o))','match');
if length(age)>1
    age1=regexpi(A,'(\d{1,3}|[a-zA-Z-]*\s[a-zA-Z-]+)(?=[-\s]*y[ea]{0,2}rs?[-\s]{1,2}↙
old)','match');
    if length(age1)==1
        age=age1;
    end
end
if ~isempty(age)
    if isnan(str2double(age))
        age1=[];
        for i=1:length(age)
            age1=[age1,words2num(age(i))];
        end
        age=age1;
    else
        age=str2double(age);
    end
    age=age(~isnan(age));
    if length(age)~=1
        if and(sum(diff(age))==0,~isempty(age))
```

```
            age=age(1);
        else
            age=missing;
        end
    end
%days old
elseif ~isempty(regexpi(A,'(\d{1,3}|[a-zA-Z-]*\s[a-zA-Z-]+)(?=[-\s]*days?[-\s]{1,2}↙
old)'))
    age=regexpi(A,'(\d{1,3}|[a-zA-Z-]*\s[a-zA-Z-]+)(?=[-\s]days?[-\s]{1,2}old)','match');
    if isnan(str2double(age))
        age1=[];
        for i=1:length(age)
            age1=[age1,words2num(age(i))];
        end
        age=age1/365;
    else
        age=str2double(age)/365;
    end
    age=age(~isnan(age));
    if length(age)~=1
        if and(sum(diff(age))==0,~isempty(age))
            age=age(1);
        else
            age=missing;
        end
    end
%weeks old
elseif ~isempty(regexpi(A,'(\d{1,3}|[a-zA-Z-]*\s[a-zA-Z-]+)(?=[-\s]*weeks?[-\s]{1,2}↙
old)'))
    age=regexpi(A,'(\d{1,3}|[a-zA-Z-]*\s[a-zA-Z-]+)(?=[-\s]weeks?[-\s]{1,2}↙
old)','match');
    if isnan(str2double(age))
        age1=[];
        for i=1:length(age)
            age1=[age1,words2num(age(i))];
        end
        age=age1/52;
    else
        age=str2double(age)/52;
    end
    age=age(~isnan(age));
    if length(age)~=1
        if and(sum(diff(age))==0,~isempty(age))
            age=age(1);
        else
            age=missing;
        end
    end
%months old
elseif ~isempty(regexpi(A,'(\d{1,3}|[a-zA-Z-]*\s[a-zA-Z-]+)(?=[-\s]*months?[-\s]{1,2}↙
old)'))
```

```
    age=regexpi(A,'(\d{1,3}|[a-zA-Z-]*\s[a-zA-Z-]+)(?=[-\s]months?[-\s]{1,2}↙
old)','match');
    if isnan(str2double(age))
        age1=[];
        for i=1:length(age)
            age1=[age1,words2num(age(i))];
        end
        age=age1/12;
    else
        age=str2double(age)/12;
    end
    age=age(~isnan(age));
    if length(age)~=1
        if and(sum(diff(age))==0,~isempty(age))
            age=age(1);
        else
            age=missing;
        end
    end
else
    age=missing;
end
%Medication Check
%Check for header
more_flag=0;
medication=regexpi(A,'MEDIC[a-z :,]+\.','match');
if strlength(medication)>1
    c=find(contains(medication,':'));
    if length(c)~=1
        medication=[];
        more_flag=1;
    else
        medication=medication(c);
    end
end
if isempty(medication)
    medication=regexpi(A,'(?-s).*\s?[,\d]+\s?m(g|l)','match');
    if and(isempty(medication),more_flag)
        medication=missing;
    elseif isempty(medication)
        medication="none";
    else
        medication=join(medication,',');
    end
end
%gender
if ~isempty(regexpi(A,'[^fe]male|[^o]man|boy'))
    sex=categorical("male");
elseif ~isempty(regexpi(A,'female|woman|girl|lady'))
    sex=categorical("female");
else
```

```
    sex=missing;
end
%flag
flag=regexpi(A,'((?<!no )brain injury)|((?<!no )TBI)|((?<!no )concussion)|((?<!no )brain↙
trauma)|((?<!no )head injury)|((?<!no )head trauma)','match');
if ~isempty(flag)
    flag=flag(1); %take only the first match
    evidence=string(regexp(A,'[:\w\'',.(\s-]+'+flag+'[\w ,)\-]*.','match'));
    evidence=join(evidence);
else
    flag="";
    evidence="";
end
method=regexpi(A,'INTRODUCTION[\w :,-.]+','match');
if isempty(method)
    method=regexpi(A,'EEG[\w :,-.]+','match');
    if isempty(method)
        method=regexpi(A,'RECORDING[\w :,-.]+','match');
        if isempty(method)
            method=missing;
        end
    end
end
if ~ismissing(method)
    method=join(method);
end
if isempty(evidence) || isempty(flag) || isempty(age) || isempty(sex) || isempty↙
(medication)
    evidence=missing;
end
notes=string(A);
end
```