```
function y = EXspectrogram(file)
% ### EXspectrogram.m ###         10.27.14
% Reads in wav file created via separate program (e.g., Audacity) and
makes a spectrogram
% NOTE: make sure sample rate specified here matches that used when
recording the data!

% -------
P.SR= 44100;  % SR data collected at [Hz]
P.windowL= 2048;   % length of window segment for FFT {2048}
P.overlap= 0.95;    % fractional overlap between window, from 0 to 1
{0.8}
P.maxF= 8000;   % max. freq. for spectrogram [Hz] {8000}
fileN= './spectrogramIS.jpg';    % filename to save image to
% -------
pts= round(P.windowL*P.overlap);      % convert fractional overlap to
# of points
disp(sprintf('Assumed sample rate = %g kHz', P.SR/1000));
%A= wavread(file);
A= audioread(file);

figure(1); clf;
spectrogram(A,blackman(P.windowL),pts,P.windowL,P.SR,'yaxis');  %
create spectrogram and plot (via built-in function)
%spectrogram(A,ones(P.windowL,1),pts,P.windowL,P.SR,'yaxis');   % in
case blackman.m is not accessible (i.e., no windowing)
axis([0 size(A,1)/P.SR 0 P.maxF])
hcb= colorbar;
xlabel('Time [s]'); ylabel('Frequency [Hz]'); ylabel(hcb,'Amplitude
[dB]');
% -------
% save picture to file as a jpg w/ a user-specified resolution
REZ= '-r180';     % resolution for exporting colormaps to jpg
print('-djpeg',REZ,[fileN]);

% -------
% also plot time waveform?
if 1==1
    t=[0:1/P.SR:(numel(A)-1)/P.SR]; % make array of time values
    figure(2); clf;
    plot(t,A,'k-');
    xlabel('Time [s]'); ylabel('Pressure [arb]'); title('time
waveform')
end

% NOTE: to play back the audio, type:
% > sound(A,SR)
% where SR is the appropriate sample rate (e.g., fiddle with if you
want to
% change the pitch)
```

```
% NOTE: To save an array (A) to .wav file, type:
% > wavwrite(A,SR,16,filename);
```