```
function [ volumeShift, pass, handles ] = volumeShift( dataSample, minReq0kph, handles )

%Initialize cells
meanDataSample = cell( 1, 5 );
normalizedDataSample = cell( 1, 5 );
bandSum = zeros( 1, 5 );
shift = zeros( 1, 5 );
pass = zeros( 1, 5 );

% Normalize levels according to S7.6.2
% For each critical operating scenario, normalize the levels of the 13
% one-third octave bands by subtracting the corresponding minimum SPL
% values specified in Table 1 for the stationary operating condition from
% each of the one-third octave band averages calculated in S7.6.1.

%Op Conditions
% 1 - Reverse
% 2 - Stationary
% 3 - 10 km/h
% 4 - 20 km/h
% 5 - 30 km/hfor ink = 1 : 5  % looping through op types

for ink = 1 : 5
    % Does not appear to be a need for the mean, it does not do anything, but does not hurt anything other than being an extra operation. Leaving it in for now.
    meanDataSample{ ink }= round( 10 * mean( dataSample{ ink }, 1 ) ) / 10;
    normalizedDataSample{ ink } = meanDataSample{ ink } - minReq0kph;
    inBetween = 10 .^ ( normalizedDataSample{ ink } ./ 10 );
    % rounding intermediate values are not recommended or required
    % inBetween = round( 10 * inBetween ) / 10; % Round to 1 decimal point
    bandSum( ink ) = 10 * log10( sum( inBetween, 2 ) );
    bandSum = round( 10 * bandSum ) / 10; % Round to 1 decimal point
end

for ink = 2 : 4
    shift( ink ) = bandSum( ink + 1 ) - bandSum( ink );
    if  shift( ink ) >= 3
        pass( ink ) = 1;
    end
end

if sum( pass ) == 3
    volumeShift = 1;
else
    volumeShift = 0;
end

handles.meanDataSample = meanDataSample ;
handles.RelVolMinReq = minReq0kph ;
handles.normalizedDataSample = normalizedDataSample ;
```

```
handles.normalizedBandSum = bandSum ;
handles.shift = shift ;
handles.pass = pass ;
```