Xcode:

```
function data = readRlrMonthly(directoryName)
% READRLRMONTHLY Reads RLR monthly data directory obtained from PSMSL
% DATA = READRLRMONTHLY(DIRECTORY) loads the contents of the unzipped rlr
% monthly data file DIRECTORY into DATA, where DATA is a structure with one
% element for each station in DIRECTORY, and has the following fields:
%   METADATA FIELDS
%   id              PSMSL id of station
%   latitude        Latitude of station
%   longitude       Longitude of station
%   name            Name of station
%   coastline       Old coastline code of station
%   stationcode     Old stationcode of station
%      (So, coastline/stationcode is the old PSMSL id of the station)
%   stationflag     Is entire station flagged for attention? True
%                   indicates yes: refer to station documentation for
%                   further information
%   DATA FIELDS
%   year            Year of data
%   month           Month of data
%   time            Time of data expressed as a Matlab date number (type
%                   HELP DATENUM for further information).
%                  Note the time given is slightly different from that
%                  given in the original file in order to more
%                  accurately represent the time observed. The value
%                  given here is the midpoint of the month. So, for
%                  example, the value given for March 1975 is:
```

```
%                   (datenum(1975,3,1) + datenum(1975,4,1)) / 2
%                 which is midday on the 16th March.
%   height          Annual height relative to RLR, in millimetres
%                   NaN indicates that data is missing
%   missing          Number of missing days of data in given month
%   dataflag         Quality control flag - true values indicate problems
%                   with the data - see station documentation for
%                   further information
%   isMtl           Flag where true value indicates that the measurement
%                 is based on mean tide level (MTL) data. Consult the
%                 psmsl.hel file for further details.
%
%   EXAMPLES OF USE:
%
%   To load the data contained in the unzipped directory C:\psmsl, enter
%      data = readRlrMonthly('C:\psmsl');
%   Alternatively, type
%      data = readRlrMonthly;
%   and navigate to the required directory
%
%   Find the index of the data for San Francisco, USA:
%      k = find(strcmp({data.name},'SAN FRANCISCO'));
%   Display the data structure for San Francisco
%     disp(data(k));
%   Display the time series of year and height
%     disp([data(k).year data(k).month data(k).height])
%
```

```
%   Find and plot data for PSMSL id 202, (Newlyn, UK)
%       k = find([data.id]==202);
%       plot(data(k).time,data(k).height);
%       datetick('x');
%
%   Plot the location of all stations:
%       plot([data.longitude],[data.latitude],'b.');
%
%   UPDATES:
%
%   2023-08-01
%   Fixed time variable, now properly reports time as centre of month as
%   described in documentation - previously provided a date half a day too
%   early.
%   Fixed dataflag for data given as MTL - these are no longer
%   automatically flagged as problematic, but use the added isMtl variable
%   instead.
%   Added isMtl variable



%   If called without input arguments, select directory
if nargin == 0
    directoryName = uigetdir;
    if directoryName == 0
        %   Aborted selection of directory, return empty array
        data = [];
        return
    end
end

%   Check directory exists
if ~exist(directoryName,'dir')
    error(['Cannot find directory ',directoryName])
end

%   Check data directory exists within this directory
if ~exist(fullfile(directoryName,'data'),'dir')
    errorString = ['Cannot find data directory ',...
        fullfile(directoryName,'data'),char(10),...
        'Please ensure the selected directory is the unzipped ',...
        'directory, not the data directory within it'];
    error(errorString)
end

%   Load catalogue file
catFile = fullfile(directoryName,'filelist.txt');
fid = fopen(catFile);
if fid == -1
    error(['Could not find catalogue file ',catFile])
end
txt = textscan(fid,'%5n;%11.6f;%12.6f; %40c; %3c; %3c; %1c');
fclose(fid);
catalogue.id = txt{1};
catalogue.latitude = txt{2};
catalogue.longitude = txt{3};
catalogue.name = txt{4};
catalogue.coastline = str2double(cellstr(txt{5}));
catalogue.stationcode = str2double(cellstr(txt{6}));
catalogue.flag = txt{7};
%   Check length of each of these fields is the same
if length(unique(cellfun(@length,txt)))~=1
    error('Error reading catalogue file')
end

%   First, get list of files to load
fileList = dir(fullfile(directoryName,'data','*.rlrdata'));
fileList = {fileList.name}';

noStations = length(fileList);
%   Check that's the same length as the number of components in the
%   catalogue
if noStations ~= length(catalogue.id)
    error(['Number of data files does not match ',...
        'the number of files in the catalogue'])
end

%   Preallocate data structure
data = struct('id',[],'latitude',[],'longitude',[],...
    'name',[],'coastline',[],'stationcode',[],'stationflag',[],...
    'year',[],'month',[],'time',[],'height',[],'missing',[],...
    'dataflag',[],'isMtl',[]);
```

```
data(noStations).id = [];

for i = 1:noStations
    %   Fill in metadata from catalogue
    thisId = str2double(strtok(fileList{i},'.'));
    k = find(catalogue.id == thisId);
    data(i).id = catalogue.id(k);
    data(i).latitude = catalogue.latitude(k);
    data(i).longitude = catalogue.longitude(k);
    data(i).name = strtrim(catalogue.name(k,:));
    data(i).coastline = catalogue.coastline(k);
    data(i).stationcode = catalogue.stationcode(k);
    data(i).stationflag = strcmp(catalogue.flag(k),'Y');

    %   Read file, and fill in elements
    thisFileName = fullfile(directoryName,'data',fileList{i});
    fid = fopen(thisFileName);
    txt = textscan(fid,'%11.4f;%6n;%2n;%1n%1n%1n');
    fclose(fid);
    %   Check all components have the same length
    if length(unique(cellfun(@length,txt)))~=1
        error(['Error reading file ',thisFileName])
    end
    %   Convert to a date string
    %   Extract month and year
    data(i).year = floor(txt{1});
    data(i).month = (round((txt{1}-data(i).year)*24)+1)/2;
    data(i).time = (datenum(data(i).year,data(i).month,1) + ...
        datenum(data(i).year,data(i).month+1,1)) / 2;
    data(i).height = txt{2};
    data(i).missing = txt{3};
    %   First column of data flags is unused.
    data(i).isMtl = txt{5}==1;
    data(i).dataflag = txt{6}==1;
    %   In height, mark missing values as Not-a-Number
    data(i).height(data(i).height==-99999) = NaN;
    %   In missing, mark interpolated data as Not-a-Number
    data(i).missing(data(i).missing==99) = NaN;
end
```