```
classdef Data
    % Class for storing and working with OpenSim data.
    %   Easy access to reading and writing of data files in the
 correct
    %   format to be used within OpenSim. Methods for data handling
    %   including subsampling, ensuring time syncronisation of various
 data
    %   inputs, etc.

    properties
        DataArray
        Values
        Labels
        Header
        isTimeSeries = false
        isConstantFrequency
        Frequency
    end

    methods
        % Construct Data object from filename.
        function obj = Data(filename)
            if nargin ~= 1
                error('Incorrect number of input arguments.')
            end
            if isstr(filename)
                obj.DataArray = importdata(filename);
                if class(obj.DataArray) == 'struct'
                    obj.Header = obj.DataArray.textdata(1:end-1,:);
                    obj.Labels = obj.DataArray.colheaders;
                    obj.Values = obj.DataArray.data;
                elseif class(obj.DataArray) == 'double'
                    obj.Values = obj.DataArray;
                else
                    error('Unrecognised data file format.')
                end
            else
                error('Error in construction: expected input filename
 as string, got %s.', class(filename)')
            end
            if sum(strcmp('time',lower(obj.Labels))) == 1
                obj.isTimeSeries = true;
            elseif sum(strcmp('time',lower(obj.Labels))) > 1
                error('More than one column recognised as time data.
 Check column labels in data file.')
            end
        end

        % Verify that the frequency at which the data is presented is
        % constant and if so store the data frequency.
        function CheckFrequency(obj)
            if obj.isTimeSeries == 0
```

```
                error('Can"t check frequency because data is not a
 time series.')
            else
                % Find the time column.
                [value, location] = max(strcmp('time', obj.Labels));
                frequency_average = sum(obj.Values(2:end,location) -
 obj.Values()/size(obj.Values,1);
            end

        end

        % Subsample data.
        f

    end

end
```

```
Error using dbstatus
Error: File: C:\Users\Daniel\Desktop\Automating Inverse Model\Data.m
 Line: 53 Column: 101
Unbalanced or unexpected parenthesis or bracket.
```

Published with MATLAB® R2016a