```
function [mobility,complexity] = HjorthParameters(xV)
% [mobility,complexity] = HjorthParameters(xV)
% HJORTHPARAMETERS computes the Hjorth parameters mobility and complexity.
% INPUTS:
% - xV           : The given scalar time series (vector of size n x 1).
% OUTPUTS
% - mobility
%========================================================================
%     <HjorthParameters.m>, v 1.0 2010/02/11 22:09:14  Kugiumtzis & Tsimpiris
%     This is part of the MATS-Toolkit http://eeganalysis.web.auth.gr/

%========================================================================
% Copyright (C) 2010 by Dimitris Kugiumtzis and Alkiviadis Tsimpiris
%                       <dkugiu@gen.auth.gr>

%========================================================================
% Version: 1.0

% LICENSE:
%     This program is free software; you can redistribute it and/or modify
%     it under the terms of the GNU General Public License as published by
%     the Free Software Foundation; either version 3 of the License, or
%     any later version.
%
%     This program is distributed in the hope that it will be useful,
%     but WITHOUT ANY WARRANTY; without even the implied warranty of
%     MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
%     GNU General Public License for more details.
%
%     You should have received a copy of the GNU General Public License
%     along with this program. If not, see http://www.gnu.org/licenses/>.

%========================================================================
% Reference : D. Kugiumtzis and A. Tsimpiris, "Measures of Analysis of Time Series (MATS):
%             A Matlab  Toolkit for Computation of Multiple Measures on Time Series Data Bases",
%             Journal of Statistical Software, in press, 2010

% Link      : http://eeganalysis.web.auth.gr/
%========================================================================
n = length(xV);
dxV = diff([0;xV]);
ddxV = diff([0;dxV]);
mx2 = mean(xV.^2);
mdx2 = mean(dxV.^2);
mddx2 = mean(ddxV.^2);

mob = mdx2 / mx2;
complexity = sqrt(mddx2 / mdx2 - mob);
mobility = sqrt(mob);
```