Method Description

General Information

Type of Entry (*Academic, Practitioner, Researcher, Student*)	Researcher
First Name	Grzegorz
Last Name	Dudek
Country	Poland
Type of Affiliation (*University, Company-Organization, Individual*)	University
Affiliation	Department of Electrical Engineering, Czestochowa University of Technology, Czestochowa, Poland

Team Members (*if applicable*):

1st Member	
First Name	
Last Name	
Country	
Affiliation	
2nd Member	
First Name	
Last Name	
Country	
Affiliation	

Information about the method utilized

Name of Method	Bunch linear extrapolation
Type of Method (*Statistical, Machine Learning, Combination, Other*)	Other
Short Description (up to 200 words)	Bunch linear extrapolation generates a set of lines. Each line passes through the two points of time series: (t_i, y_i) and (t_n, y_n), where (t_n, y_n) is the last point of time series, and (t_i, y_i) is the i-th point, i = 1, 2, ..., n-1. So, as a result we have n-1 lines, all passing through the last n-th point. Then, we determine the median line for these n-1 lines. The forecasts for the next h points are the median for (n+1, n+2, ..., n+h). Having the median line we calculate 95% prediction interval using standard approach for a linear model.

Extended Description:

For yearly, monthly, weekly and daily time series the model is constructed as follows:

1. If the number of points in the time series n>10*h (h is a forecast horizon) then remove first points from 1 to n-10*h. So, the time series is shorten to 10*h last points.
2. Create n-1 lines passing through the two points of time series: (t_i, y_i) and (t_n, y_n), where (t_n, y_n) is the last point of the time series, and (t_i, y_i) is the i-th point, i = 1, 2, ..., n-1. So, as a result we have n-1 lines, all passing through the last n-th point.
3. Determine the median line for a bunch of n-1 lines created in step 2. Each point of the median line is calculated as a median of corresponding points from n-1 lines.
4. Determine the forecasts for the next h points as the median for (n+1, n+2, ..., n+h).
5. Calculate 95% prediction interval using standard approach for a linear model:

$$\hat{y} \pm t_{(1-\propto/2,n-2)}\, s_y \sqrt{1+\frac{1}{n}+\frac{(t^{*}-\bar{t})^2}{(n-1)s_t^2}}$$

For quartely and hourly time series expressing seasonal patterns the seasonal version of the model is used. In this case the quartely time series is decomposed on four time series:

1) {y_t}, t = 1, 5, 9, ...,
2) {y_t}, t = 2, 6, 10, ...,
3) {y_t}, t = 3, 7, 11, ...,
4) {y_t}, t = 4, 8, 12, ...

Similarly the hourly time series is decomposed into 24 time series:

1) {y_t}, t = 1, 25, 49, ...,
2) {y_t}, t = 2, 26, 50, ...,

...

24) {y_t}, t = 24, 48, 72, ...

Each of these new time series is forecasted independently.

The model is constructed as follows:

1. If the number of points in the time series n>10*h (h is a forecast horizon) then remove first points from 1 to n-10*h. So, the time series is shorten to 10*h last points.
2. Decompose the time series into 4 or 24 time series (see above).
3. For each decomposed time series having n* points create n*-1 lines passing through the two points of time series: (t_i, y_i) and (t_{n^*}, y_{n^*}), where (t_{n^*}, y_{n^*}) is the last point of time series, and (t_i, y_i) is the i-th point, i = 1, 2, ..., n*-1. So, as a result we have n*-1 lines, all passing through the last n*-th point.
4. For each decomposed time series determine the median line for a bunch of n*-1 lines created in step 2. Each point of the median line is calculated as a median of the corresponding points from n*-1 lines.

5. For each decomposed time series determine the forecasts for the next h* points as the median for (n*+1, n*+2, ..., n*+h*), where in our case h* = 2.
6. For each decomposed time series calculate 95% prediction interval using standard approach for a linear model:

$$\hat{y} \pm t_{(1-\propto/2,n-2)}\, s_y \sqrt{1+\frac{1}{n^*}+\frac{(t^*-\bar{t})^2}{(n^*-1)s_t^2}}$$

BLE model features:
- no parameters
- no assumptions
- no initialization
- no training
- no complex calculations
- clear and understandable model
- simple implementation in any environment
- fast execution
- seasonal approach needs time series decomposition

Source and output files

The BLE models are implemented in Matlab:
- BLE_for_YeMoWeDa.m - model for yearly, monthly, weekly and daily time series,
- BLE_for_QuHo.m - model for seasonal quartely and hourly time series.

The input data are provided in the files:
- Yearly-train1.csv
- Quarterly-train1.csv
- Monthly-train1.csv
- Weekly-train1.csv
- Daily-train1.csv
- Hourly-train1.csv

which are the same as original data files but without quotation marks around numerical values.

To execute the m-file select the input file in the source code. Results are saved in output files (variable 'file'):
- Yerly_wyn.mat
- Quart_wyn.mat
- Month_wyn.mat
- Week_wyn.mat
- Dail_wyn.mat
- Hourl_wyn.mat

Each file include: time series label, forecasts (ypro), lower (y025) and upper (y975) bounds of prediction intervals.

Remark: the scripts for running need Statistics and Machine Learning Toolbox to be installed. It is needed for prediction intervals calculation where tinv function is used (Student's t inverse cumulative distribution function) which is from Statistics and Machine Learning Toolbox.