A MATLAB LIBRARY OF TEMPORAL DISAGGREGATION METHODS: SUMMARY

Enrique M. Quilis

Instituto Nacional de Estadística
Paseo de la Castellana, 183
28046 - Madrid (SPAIN)

December, 2002

1 The author thanks Ana Abad, Juan Bógalo and Silvia Retlono for their help.
1. INTRODUCTION

The library includes a set of functions to perform temporal disaggregation (distribution, averaging and interpolation), according to the following structure:

Adjustment or quadratic programming methods:

- bfl
- denton_uni

  served by: tduni_print (ASCII output), tduni_plot (graphic output)

Model-based (or BLUE) methods:

- chowlin
- fernandez
- litterman

  served by: td_print (ASCII output), td_plot (graphic output)

Multivariate methods that include a transversal restriction:

- denton
- difonzo

  served by: mtd_print (ASCII output), mtd_plot (graphic output)

Extrapolation is feasible using chowlin, fernandez, litterman and difonzo. Constrained extrapolation can be performed also by means of difonzo.

The presentation of the functions is self-contained: help text, script to run the function and output (ASCII file and plots).

Matlab source code is freely available if requested at: info@ine.es
2. BOOT-FEIBES-LISMAN

PURPOSE: Temporal disaggregation using the Boot-Feibes-Lisman method

SYNTAX: res=bfl(Y,ta,d,s);

OUTPUT: res: a structure
    res.meth = 'Boot-Feibes-Lisman';
    res.N = Number of low frequency data
    res.ta = Type of disaggregation
    res.s = Frequency conversion
    res.d = Degree of differencing
    res.y = High frequency estimate
    res.et = Elapsed time

INPUT: Y: Nx1 ---> vector of low frequency data
    ta: type of disaggregation
        ta=1 ---> sum (flow)
        ta=2 ---> average (index)
        ta=3 ---> last element (stock) ---> interpolation
    d: objective function to be minimized: volatility of ...
        d=0 ---> levels
        d=1 ---> first differences
        d=2 ---> second differences
    s: number of high frequency data points for each low frequency data point
        s= 4 ---> annual to quarterly
        s=12 ---> annual to monthly
        s= 3 ---> quarterly to monthly

LIBRARY: aggreg

SEE ALSO: tduni_print, tduni_plot

"Further methods of derivation of quarterly figures from annual data",

Application:

```matlab
Y=load('c:\x\td\data\Y.anu');
res=bfl(Y,1,1,12);
tduni_print(res,'td.sal');
tduni_plot(res);
edt td.sal
```
TEMPORAL DISAGGREGATION METHOD: Boot-Feibes-Lisman

- Number of low-frequency observations: 22
- Frequency conversion: 12
- Number of high-frequency observations: 264

- Degree of differencing: 1
- Type of disaggregation: sum (flow).

High frequency series (columnwise):

4972.2800
4971.1389
..........
..........
..........
7898.7692
7899.3631
7899.6600

Elapsed time: 0.3200
3. DENTON

PURPOSE: Temporal disaggregation using the Denton method

SYNTAX: res=denton_uni(Y,x,ta,d,s);

OUTPUT: res: a structure
  res.meth = 'Denton';
  res.N     = Number of low frequency data
  res.ta    = Type of disaggregation
  res.s     = Frequency conversion
  res.d     = Degree of differencing
  res.y     = High frequency estimate
  res.U     = Low frequency residuals
  res.u     = High frequency residuals
  res.et    = Elapsed time

INPUT: Y: Nx1 ---> vector of low frequency data
  x: nx1 ---> vector of low frequency data
  ta: type of disaggregation
     ta=1 ---> sum (flow)
     ta=2 ---> average (index)
     ta=3 ---> last element (stock) ---> interpolation
  d: objective function to be minimized: volatility of ...
     d=0 ---> levels
     d=1 ---> first differences
     d=2 ---> second differences
  s: number of high frequency data points for each low frequency data point
     s= 4 ---> annual to quarterly
     s=12 ---> annual to monthly
     s= 3 ---> quarterly to monthly

LIBRARY: aggreg, bfl

SEE ALSO: tduni_plot, tduni_print


Application:

Y=load('c:\td\data\Y.prn');
x=load('c:\td\data\x.ind');
res=denton_uni(Y,x,1,1,4);
tduni_print(res,'td.sal');
tduni_plot(res);
edit td.sal
ASCII file containing detailed output:

********************************************************************************
TEMPORAL DISAGGREGATION METHOD: Denton
********************************************************************************

----------------------------------------------------
Number of low-frequency observations :   22
Frequency conversion                  :    4
Number of high-frequency observations :   88
----------------------------------------------------
Degree of differencing                :    1
Type of disaggregation: sum (flow).
----------------------------------------------------
High frequency series (columnwise):
----------------------------------------------------
15374.9285
15169.7571
..........
..........
..........
24883.3098
20609.0705
24415.4509
----------------------------------------------------
Elapsed time:   0.0500
4. CHOW-LIN

PURPOSE: Temporal disaggregation using the Chow-Lin method
-------------------------------------------------------------
SYNTAX: res=chowlin(Y,x,ta,s,type);
-------------------------------------------------------------
OUTPUT: res: a structure
    res.meth    = 'Chow-Lin';
    res.ta      = type of disaggregation
    res.type    = method of estimation
    res.N       = nobs. of low frequency data
    res.n       = nobs. of high-frequency data
    res.pred    = number of extrapolations
    res.s       = frequency conversion between low and high freq.
    res.p       = number of regressors (including intercept)
    res.Y       = low frequency data
    res.x       = high frequency indicators
    res.y       = high frequency estimate
    res.y_dt    = high frequency estimate: standard deviation
    res.y_lo    = high frequency estimate: sd - sigma
    res.y_up    = high frequency estimate: sd + sigma
    res.u       = high frequency residuals
    res.U       = low frequency residuals
    res.beta    = estimated model parameters
    res.beta_sd = estimated model parameters: standard deviation
    res.beta_t  = estimated model parameters: t ratios
    res.rho     = innovational parameter
    res.aic     = Information criterion: AIC
    res.bic     = Information criterion: BIC
    res.val     = Objective function used by the estimation method
    res.r       = grid of innovational parameters used by the estimation method
-------------------------------------------------------------
INPUT: Y: Nx1 ---> vector of low frequency data
    x: nxp ---> matrix of high frequency indicators (without intercept)
    ta: type of disaggregation
        ta=1 ---> sum (flow)
        ta=2 ---> average (index)
        ta=3 ---> last element (stock) ---> interpolation
    s: number of high frequency data points for each low frequency data points
        s= 4 ---> annual to quarterly
        s=12 ---> annual to monthly
        s= 3 ---> quarterly to monthly
    type: estimation method:
        type=0 ---> weighted least squares
        type=1 ---> maximum likelihood
-------------------------------------------------------------
LIBRARY: aggreg
-------------------------------------------------------------
SEE ALSO: litterman, fernandez, td_plot, td_print
-------------------------------------------------------------
Application:

```matlab
Y=load('c:\td\data\Y.prn');
x=load('c:\td\data\x.ind');
res=chowlin(Y,x,1,4,1);
td_print(res,'td.sal',1);      % op1=1: series are printed in ASCII file
td_plot(res);
edit td.sal
```

ASCII file containing detailed output:

```
*************************
TEMPORAL DISAGGREGATION METHOD: Chow-Lin
*************************
---------------------------------------------------------------------
Number of low-frequency observations :   22
Frequency conversion                 :    4
Number of high-frequency observations:   88
Number of extrapolations             :    0
Number of indicators (+ constant)    :    2
---------------------------------------------------------------------
Type of disaggregation: sum (flow).
---------------------------------------------------------------------
Estimation method: Maximum likelihood.
---------------------------------------------------------------------
Beta parameters (columnwise):
* Estimate
* Std. deviation
* t-ratios
---------------------------------------------------------------------
215.4518        111.7079          1.9287
0.9828          0.0069        142.0272
---------------------------------------------------------------------
Innovational parameter:   0.7600
---------------------------------------------------------------------
AIC:  10.0340
BIC:  10.1828
---------------------------------------------------------------------
Low-frequency correlation
 - levels     : 0.9998
 - yoy rates  : 0.9617
---------------------------------------------------------------------
High-frequency correlation
 - levels     : 0.9998
 - yoy rates  : 0.9812
---------------------------------------------------------------------
High-frequency volatility of yoy rates
 - estimate   : 8.4282
 - indicator  : 9.0226
 - ratio      : 0.9341
---------------------------------------------------------------------
```
High frequency series (columnwise):
* Estimate
* Std. deviation
* 1 sigma lower limit
* 1 sigma upper limit
* Residuals

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5400.9896</td>
<td>114.8247</td>
<td>5286.1649</td>
<td>5515.8143</td>
<td>112.3095</td>
</tr>
<tr>
<td>5311.2409</td>
<td>83.7296</td>
<td>5227.5112</td>
<td>5394.9705</td>
<td>128.7034</td>
</tr>
<tr>
<td>........</td>
<td>.......</td>
<td>........</td>
<td>.........</td>
<td>..........</td>
</tr>
<tr>
<td>........</td>
<td>.......</td>
<td>........</td>
<td>.........</td>
<td>..........</td>
</tr>
<tr>
<td>30079.6885</td>
<td>86.7557</td>
<td>29992.9328</td>
<td>30166.4443</td>
<td>-97.4913</td>
</tr>
<tr>
<td>25874.7702</td>
<td>86.2867</td>
<td>25788.4835</td>
<td>25961.0569</td>
<td>-43.9249</td>
</tr>
<tr>
<td>29614.4998</td>
<td>116.3242</td>
<td>29498.1756</td>
<td>29730.8240</td>
<td>-16.2417</td>
</tr>
</tbody>
</table>

Elapsed time: 1.8100
A variant to be applied with a fixed innovational parameter:

**PURPOSE:** Temporal disaggregation using the Chow-Lin method
rho parameter is fixed (supplied by the user)

**SYNTAX:** res=chowlin_fix(Y,x,ta,s,type,rho);
5. FERNÁNDEZ

PURPOSE: Temporal disaggregation using the Fernandez method
-------------------------------------------------------------
SYNTAX: res=fernandez(Y,x,ta,s);
-------------------------------------------------------------
OUTPUT: res: a structure
    res.meth    =$'Fernandez' ;
    res.ta      = type of disaggregation
    res.type    = method of estimation
    res.N       = nobs. of low frequency data
    res.n       = nobs. of high-frequency data
    res.pred    = number of extrapolations
    res.s       = frequency conversion between low and high freq.
    res.p       = number of regressors (including intercept)
    res.Y       = low frequency data
    res.x       = high frequency indicators
    res.y       = high frequency estimate
    res.y_dt    = high frequency estimate: standard deviation
    res.y_lo    = high frequency estimate: sd - sigma
    res.y_up    = high frequency estimate: sd + sigma
    res.u       = high frequency residuals
    res.U       = low frequency residuals
    res.beta    = estimated model parameters
    res.beta_sd = estimated model parameters: standard deviation
    res.beta_t  = estimated model parameters: t ratios
    res.aic     = Information criterion: AIC
    res.bic     = Information criterion: BIC
-------------------------------------------------------------
INPUT: Y: Nx1 ---> vector of low frequency data
    x: nxp ---> matrix of high frequency indicators (without intercept)
    ta: type of disaggregation
        ta=1 ---> sum (flow)
        ta=2 ---> average (index)
        ta=3 ---> last element (stock) ---> interpolation
    s: number of high frequency data points for each low frequency data points
    s= 4 ---> annual to quarterly
    s=12 ---> annual to monthly
    s= 3 ---> quarterly to monthly
-------------------------------------------------------------
LIBRARY: aggreg
-------------------------------------------------------------
SEE ALSO: chowlin, litterman, td_plot, td_print
-------------------------------------------------------------

Application:
```
Y=load('c:\xtd\data\Y.prn');
x=load('c:\xtd\data\x.tri');
res=fernandez(Y,x,1,4);

td_print(res,'td.sal',1);      % op1=1: series are printed in ASCII file
td_plot(res);                 
edit td.sal
```

**ASCII file containing detailed output:**

```
****************************************************
TEMPORAL DISAGGREGATION METHOD: Fernandez
****************************************************
----------------------------------------------------
Number of low-frequency observations :   22
Frequency conversion                 :    4
Number of high-frequency observations:   90
Number of extrapolations             :    2
Number of indicators (+ constant)    :    2
----------------------------------------------------
Type of disaggregation: sum (flow).
----------------------------------------------------
Estimation method: Maximum likelihood.
----------------------------------------------------
Beta parameters (columnwise):
  * Estimate
  * Std. deviation
  * t-ratios

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>564.9834</td>
<td>195.9404</td>
<td>2.8834</td>
</tr>
<tr>
<td>0.9360</td>
<td>0.0292</td>
<td>32.0284</td>
</tr>
</tbody>
</table>

Innovational parameter:   1.0000
----------------------------------------------------
AIC:   9.6079
BIC:   9.7567
----------------------------------------------------
Low-frequency correlation
  - levels     : 0.9998
  - yoy rates  : 0.9617
----------------------------------------------------
High-frequency correlation
  - levels     : 0.9997
  - yoy rates  : 0.9817
----------------------------------------------------
High-frequency volatility of yoy rates
  - estimate   : 8.3477
  - indicator  : 9.1506
  - ratio      : 0.9123
```
High frequency series (columnwise):
* Estimate
* Std. deviation
* 1 sigma lower limit
* 1 sigma upper limit
* Residuals

<table>
<thead>
<tr>
<th></th>
<th>Estimate</th>
<th>Std. deviation</th>
<th>1 sigma lower limit</th>
<th>1 sigma upper limit</th>
<th>Residuals</th>
</tr>
</thead>
<tbody>
<tr>
<td>5396.6742</td>
<td>91.6250</td>
<td>5305.0492</td>
<td>5488.2992</td>
<td>-0.0000</td>
<td></td>
</tr>
<tr>
<td>5297.9198</td>
<td>60.8871</td>
<td>5237.0327</td>
<td>5358.8069</td>
<td>2.3349</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30021.1833</td>
<td>73.6977</td>
<td>29947.4856</td>
<td>30094.8810</td>
<td>920.9566</td>
<td></td>
</tr>
<tr>
<td>26022.3844</td>
<td>108.3992</td>
<td>25913.9852</td>
<td>26130.7837</td>
<td>977.8951</td>
<td></td>
</tr>
<tr>
<td>29586.1687</td>
<td>92.9937</td>
<td>29493.1750</td>
<td>29679.1625</td>
<td>1006.3644</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28366.5459</td>
<td>140.8431</td>
<td>28225.7028</td>
<td>28507.3889</td>
<td>1006.3644</td>
<td></td>
</tr>
<tr>
<td>29461.6792</td>
<td>176.5235</td>
<td>29286.1557</td>
<td>29638.2027</td>
<td>1006.3644</td>
<td></td>
</tr>
</tbody>
</table>

Elapsed time: 0.0500

Graphs are the same than in the Chow-Lin case, except that the first one (objective function vs innovational parameter) is not generated.
6. LITTERMAN

PURPOSE: Temporal disaggregation using the Litterman method
------------------------------------------------------------
SYNTAX: res=litterman(Y,x,ta,s,type);
------------------------------------------------------------
OUTPUT: res: a structure
  res.meth = 'Litterman';
  res.ta    = type of disaggregation
  res.type  = method of estimation
  res.N     = nobs. of low frequency data
  res.n     = nobs. of high-frequency data
  res.pred  = number of extrapolations
  res.s     = frequency conversion between low and high freq.
  res.p     = number of regressors (including intercept)
  res.Y     = low frequency data
  res.x     = high frequency indicators
  res.y     = high frequency estimate
  res.y_dt  = high frequency estimate: standard deviation
  res.y_lo  = high frequency estimate: sd - sigma
  res.y_up  = high frequency estimate: sd + sigma
  res.u     = high frequency residuals
  res.U     = low frequency residuals
  res.beta = estimated model parameters
  res.beta_sd = estimated model parameters: standard deviation
  res.beta_t = estimated model parameters: t ratios
  res.rho   = innovational parameter
  res.aic   = Information criterion: AIC
  res.bic   = Information criterion: BIC
  res.val   = Objective function used by the estimation method
  res.r     = grid of innovational parameters used by the estimation method
------------------------------------------------------------
INPUT: Y: Nx1 ---> vector of low frequency data
  x: nxp ---> matrix of high frequency indicators (without intercept)
  ta: type of disaggregation
    ta=1 ---> sum (flow)
    ta=2 ---> average (index)
    ta=3 ---> last element (stock) ---> interpolation
  s: number of high frequency data points for each low frequency data points
    s= 4 ---> annual to quarterly
    s=12 ---> annual to monthly
    s= 3 ---> quarterly to monthly
  type: estimation method:
    type=0 ---> weighted least squares
    type=1 ---> maximum likelihood
------------------------------------------------------------
LIBRARY: aggreg
------------------------------------------------------------
SEE ALSO: chowlin, fernandez, td_plot, td_print
------------------------------------------------------------
Application:

```matlab
Y=load('c:\xtld\data\Y.prn');
x=load('c:\xtld\data\x.tri');
res=litterman(Y,x,1,4,0);
td_print(res,'td.sal',0); % op1=0: series are not printed in ASCII file
td_plot(res);
edit td.sal
```

ASCII file containing detailed output:

```
TEMPORAL DISAGGREGATION METHOD: Litterman

Number of low-frequency observations :   22
Frequency conversion                 :    4
Number of high-frequency observations:   90
Number of extrapolations             :    2
Number of indicators (+ constant)    :    2

Type of disaggregation: sum (flow).

Estimation method: Weighted least squares.

Beta parameters (columnwise):
   * Estimate
   * Std. deviation
   * t-ratios

<table>
<thead>
<tr>
<th></th>
<th>Estimate</th>
<th>Std. deviation</th>
<th>t-ratios</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1205.4851</td>
<td>233.5241</td>
<td>5.1621</td>
</tr>
<tr>
<td>2</td>
<td>0.7910</td>
<td>0.0480</td>
<td>16.4821</td>
</tr>
</tbody>
</table>

Innovational parameter: 0.9700

AIC: 7.9478
BIC: 8.0966

Low-frequency correlation
- levels : 0.9998
- yoy rates : 0.9617

High-frequency correlation
- levels : 0.9994
- yoy rates : 0.9735

High-frequency volatility of yoy rates
- estimate : 7.6249
- indicator : 9.1506
- ratio : 0.8333

Elapsed time: 2.5300
```
A variant to be applied with a fixed innovational parameter:

**PURPOSE:** Temporal disaggregation using the Litterman method
mu parameter is fixed (supplied by the user)

--------------------------------------------
**SYNTAX:** \( \text{res}=\text{litterman}(Y,x,ta,s,type,\mu) \);

Graphical output contains the same information than in the Chow-Lin case.
7. MULTIVARIATE DENTON

PURPOSE: Multivariate temporal disaggregation with transversal constraint

SYNTAX: res = denton(Y,x,z,ta,s,d);

OUTPUT: res: a structure
- res.meth = 'Multivariate Denton';
- res.N = Number of low frequency data
- res.n = Number of high frequency data
- res.pred = Number of extrapolations (=0 in this case)
- res.ta = Type of disaggregation
- res.s = Frequency conversion
- res.d = Degree of differencing
- res.y = High frequency estimate
- res.et = Elapsed time

INPUT: Y: NxM ---> M series of low frequency data with N observations
- x: nxM ---> M series of high frequency data with n observations
- z: nx1 ---> high frequency transversal constraint
- ta: type of disaggregation
  - ta=1 ---> sum (flow)
  - ta=2 ---> average (index)
  - ta=3 ---> last element (stock) ---> interpolation
- s: number of high frequency data points for each low frequency data points
  - s= 4 ----> annual to quarterly
  - s=12 ----> annual to monthly
  - s= 3 ----> quarterly to monthly
- d: objective function to be minimized: volatility of ...
  - d=0 ----> levels
  - d=1 ----> first differences
  - d=2 ----> second differences

LIBRARY: aggreg, dif, vec, desvec

SEE ALSO: difonzo, mtd_print, mtd_plot

REFERENCE: di Fonzo, T. (1994) "Temporal disaggregation of a system of
time series when the aggregate is known: optimal vs. adjustment methods",
INSEE-Eurostat Workshop on Quarterly National Accounts, Paris, december

Application:

Y=load('YY.anu'); % Loading low frequency data
x=load('x.tri'); % Loading high frequency data
z=load('z.prn'); % Loading high frequency transversal restriction
res=denton(Y,x,z,2,4,1);
mtd_print(res,'mtd.sal');
edit mtd.sal;
mtd_plot(res,z);
ASCII file containing detailed output:

******************************************************************************
TEMPORAL DISAGGREGATION METHOD: Multivariate Denton
******************************************************************************

Number of low-frequency observations : 23
Frequency conversion : 4
Number of high-frequency observations : 92
Number of extrapolations : 0

Degree of differencing : 1
Type of disaggregation: average (index).

High frequency series (columnwise):
  * Point estimate

  3752.9096  4982.6505
  3459.3681  5257.1693
  ........    ........
  ........    ........
  ........    ........
  2757.8458  8545.8074
  2825.1411  8624.4561
  2867.5816  8657.9733

Elapsed time:  0.2800
PURPOSE: Multivariate temporal disaggregation with transversal constraint
---------------------------------------------------------------------
SYNTAX: res = difonzo(Y,x,z,ta,s,type);
---------------------------------------------------------------------
OUTPUT: res: a structure
    res.meth = 'Multivariate di Fonzo';
    res.N = Number of low frequency data
    res.n = Number of high frequency data
    res.pred = Number of extrapolations
    res.ta = Type of disaggregation
    res.s = Frequency conversion
    res.type = Model for high frequency innovations
    res.beta = Model parameters
    res.y = High frequency estimate
    res.d_y = High frequency estimate: std. deviation
    res.et = Elapsed time
---------------------------------------------------------------------
INPUT: Y: NxM ---> M series of low frequency data with N observations
    x: nxM ---> M series of high frequency data with n observations
    z: nx1 ---> high frequency transversal constraint with nz obs.
    ta: type of disaggregation
        ta=1 ---> sum (flow)
        ta=2 ---> average (index)
        ta=3 ---> last element (stock) ---> interpolation
    s: number of high frequency data points for each low frequency data points
        s= 4 ---> annual to quarterly
        s=12 ---> annual to monthly
        s= 3 ---> quarterly to monthly
    type: model for the high frequency innovations
        type=0 ---> multivariate white noise
        type=1 ---> multivariate random walk
---------------------------------------------------------------------
NOTE: Extrapolation is automatically performed when n>sN.
    If n=nz>sN restricted extrapolation is applied.
    Finally, if n>nz>sN extrapolation is performed in constrained
    form in the first nz-sN observations and in free form in
    the last n-nz observations.
---------------------------------------------------------------------
LIBRARY: aggreg, dif, vec, desvec
---------------------------------------------------------------------
SEE ALSO: denton, mtd_print, mtd_plot
---------------------------------------------------------------------
REFERENCE: di Fonzo, T.(1990)"The estimation of M disaggregate time
    series when contemporaneous and temporal aggregates are known", Review
Application:

```matlab
Y=load('YY.anu'); % Loading low frequency data
x=load('x.tri'); % Loading high frequency data
z=load('z.prn'); % Loading high frequency transversal restriction
res = difonzo(Y,x,z,2,4,1);
mtd_print(res,'mtd.sal');
edit mtd.sal;
mtd_plot(res,z);
```

ASCII file containing detailed output:

```
*******************************************************
TEMPORAL DISAGGREGATION METHOD: Multivariate di Fonzo
*******************************************************
-------------------------------------------------------
Number of low-frequency observations : 23
Frequency conversion                  : 4
Number of high-frequency observations : 92
Number of extrapolations             : 0
-------------------------------------------------------
Model for the innovations: random walk.
Type of disaggregation: average (index).
-------------------------------------------------------
High frequency series (columnwise):
   * Point estimate
```
```
3413.3839  5322.1762
3447.4092  5269.1282
........     ........
........     ........
........     ........
2758.4657  8545.1875
2817.9882  8631.6090
2856.1605  8669.3944
```
High frequency series (columnwise):
* Std. desviation

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>197.8732</td>
<td>197.8732</td>
</tr>
<tr>
<td>127.3900</td>
<td>127.3900</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>137.9397</td>
<td>137.9397</td>
</tr>
<tr>
<td>128.1006</td>
<td>128.1006</td>
</tr>
<tr>
<td>194.9112</td>
<td>194.9112</td>
</tr>
</tbody>
</table>

Elapsed time: 0.3300