High-Frequency Time-Series (HFTS) objects

Table of Contents

Convert Chopper use of dataset to a HFTS data object

Create the test data from FDS

```
dbstruct.username = '';
dbstruct.password = '';
dbstruct.database = 'ICM_ALGOS';
dbstruct.driver   = 'com.microsoft.sqlserver.jdbc.SQLServerDriver';
dbstruct.databaseurl = 'jdbc:sqlserver://icmjhbmsqldev03:56949;database=ICM_ALGOS;
conn = database(dbstruct.database,dbstruct.username,dbstruct.password,dbstruct.dri
% 10-min Bar data
dataA = fetchtrth(conn,{'AGLJ.J'},'Intraday 10Min',[datenum('2-Oct-2011'), datenum
dataB = fetchtrth(conn,{'BILJ.J'},'Intraday 10Min',[datenum('2-Oct-2011'), datenum
% 1-min Bar data
dataC = fetchtrth(conn,{'AGLJ.J'},'Intraday 1Min',[datenum('2-Oct-2011'), datenum(
dataD = fetchtrth(conn,{'BILJ.J'},'Intraday 1Min',[datenum('2-Oct-2011'), datenum(
% Combined 1-min bar data
dataE = fetchtrth(conn,{'AGLJ.J','BILJ.J'},'Intraday 1Min',[datenum('2-Oct-2011'),
% Trade-Sales data
dataF = fetchtrth(conn,{'AGLJ.J'},'Trade',[datenum('2-Oct-2011'), datenum('31-Jan-
dataG = fetchtrth(conn,{'BILJ.J'},'Trade',[datenum('2-Oct-2011'), datenum('31-Jan-
% Combined Tick Data
dataH = fetchtrth(conn,{'AGLJ.J','BILJ.J','SBKJ.J'},'Trade',[datenum('2-Oct-2011')
```

Create test data directly from TRTH

```
try
```

```
    %  connection objects
    r = rdth('tim.gebbie@investec.co.za','blackbdyrad220');
    % The reduced basket RIC codes and request type
    Tickers = {'AGL,BIL,SBK'};
    Tickers = tick2tick(commalist2cell(Tickers{:}),'RIC','JSE');
    for i=1:length(Tickers),
        Exchange{i} = 'JNB';
        Domain{i} = 'EQU';
    end;
    reqtype = 'TimeAndSales';
    messtype = 'Trade';
    tradefields = {'Price','Volume','Mid Price'};
    edate = busdays(today-10,today,1); % load data for 2 days prior
    sdate = edate(end-1);
    edate = edate(end-1);
    dataI = trth2struct(r,Tickers,tradefields,sdate,edate,reqtype,messtype,Exchang
catch
end
```

Create HFTS from .CSV file

Create HFTS from Reuter real-time

```
rc.session = 'myNS::SSLSession';
rc.source = 'IDN_RDF';
rc.id = 'tgebbie';
rconn = reuters(rc.session,rc.source,rc.id,[],1);
[dataM,ts0] = fetchreuters(rconn,{'AGLJ.J','BILJ.J','SBKJ.J'},{'TRDPRC_1'});
tsM = reuters2cell(ts0,{'TRDPRC_1','TRDVOL_1'},{'Price','Volume'}),
tsM.AGL
tsM = hfts(tsM);
tsM.AGL
tsM1 = mergets(tsM);
tsM1.Price
rtsM1 = resample(tsM,'m');
rtsM1 = mergets(rtsM1);
rtsM1.Price
```

```
        tsM =

            AGL: {2x5 cell}
            BIL: {2x5 cell}
            SBK: {2x5 cell}


        ans =

            'RIC'       'DateL'          'TimeL'    'Price'    'Volume'
            'AGLJ.J'    '13 APR 2012'    '12:27'    [28795]    [  2100]

        Warning: The class file for 'hfts' has been changed; but the change cannot
```

```
applied because objects based on the old class file still exist. If you us
those objects, you might get unexpected results. You can use the 'clear' c
to remove those objects. See 'help clear' for information on how to remove
objects.

ans = 

    DateTime      Price    RIC       Volume
    7.3497e+05    28795    AGLJ.J    2100


ans = 

    DateTime      AGL      BIL      SBK
    7.3497e+05    28795    24115    11190


ans = 

    DateTime      AGL      BIL      SBK
    7.3497e+05    28795    24115    11190
```

Class constructor and methods for Tick Data

```
tsH = hfts(dataH),
tsH = aggregate(tsH);
tsH.AGL(1:4,:),
size(tsH.AGL),
size(tsH.BIL),
tsH.series,
tsH0 = hfts(dataH,{'Price','Volume'}),
tsH0 = aggregate(tsH);
tsH1 = mergets(tsH),
tsH1.Price(1:2,:),
tsH2 = resample(tsH,1/3600),
tsH3 = tsH,
tsH3.freq = 's',
```

```
Warning: The class file for 'hfts' has been changed; but the change cannot
applied because objects based on the old class file still exist. If you us
those objects, you might get unexpected results. You can use the 'clear' c
to remove those objects. See 'help clear' for information on how to remove
objects.
  hfts

  Properties:
    freq: 'unknown'


 series : AGL, BIL, SBK
 fields : DateTime, MidPrice, Price, RIC, Volume
```

```
ans = 

    DateTime      MidPrice    Price    RIC       Volume
    7.3485e+05    NaN         29650    AGLJ.J    4542  
    7.3485e+05    NaN         29640    AGLJ.J     300  
    7.3485e+05    NaN         29650    AGLJ.J       5  
    7.3485e+05    NaN         29650    AGLJ.J     378  


ans =

       20277           5


ans =

       18326           5


ans = 

    'AGL'
    'BIL'
    'SBK'

Warning: The class file for 'hfts' has been changed; but the change cannot
applied because objects based on the old class file still exist. If you us
those objects, you might get unexpected results. You can use the 'clear' c
to remove those objects. See 'help clear' for information on how to remove
objects. 
  hfts

  Properties:
    freq: 'unknown'


 series : AGL, BIL, SBK
 fields : RIC, DateTime, Price, Volume
Warning: Observations with default values added to dataset variables. 
Warning: Observations with default values added to dataset variables. 
Warning: Observations with default values added to dataset variables. 
  hfts

  Properties:
    freq: 'unknown'


 series : MidPrice, Price, RIC, Volume
 fields : DateTime, AGL, BIL, SBK

ans = 

    DateTime      AGL    BIL      SBK
    7.3485e+05    NaN    23949    NaN
    7.3485e+05    NaN    23902    NaN

  hfts

  Properties:
    freq: 'uniform'


 series : AGL, BIL, SBK
 fields : DateTime, MidPrice, Price, RIC, Volume
  hfts

  Properties:
    freq: 'unknown'


 series : AGL, BIL, SBK
 fields : DateTime, MidPrice, Price, RIC, Volume
  hfts

  Properties:
    freq: 'seconds'


 series : AGL, BIL, SBK
 fields : DateTime, MidPrice, Price, RIC, Volume
```

convert to FINTS objects

```
f1 = fints(tsH);
f2 = fints(tsH1);
```

```
Warning: HFTS object freq is not MINUTES; may be incorrectly aggregated
Warning: HFTS object freq is not MINUTES; may be incorrectly aggregated
```

Merge HFTS objects

```
tsAM = merge(tsH,tsM);
```

Test TRTH DATA

```
try
tsI = hfts(dataI,{'Price','Volume'}),
catch
end
```

```
Warning: The class file for 'hfts' has been changed; but the change cannot
applied because objects based on the old class file still exist. If you us
those objects, you might get unexpected results. You can use the 'clear' c
to remove those objects. See 'help clear' for information on how to remove
objects.
```

```
Warning: Variable names were modified to make them valid MATLAB identifier
Warning: Variable names were modified to make them valid MATLAB identifier
Warning: Variable names were modified to make them valid MATLAB identifier
  hfts

  Properties:
    freq: 'unknown'


 series : AGL, BIL, SBK
 fields : RIC, DateTime, Price, Volume
```

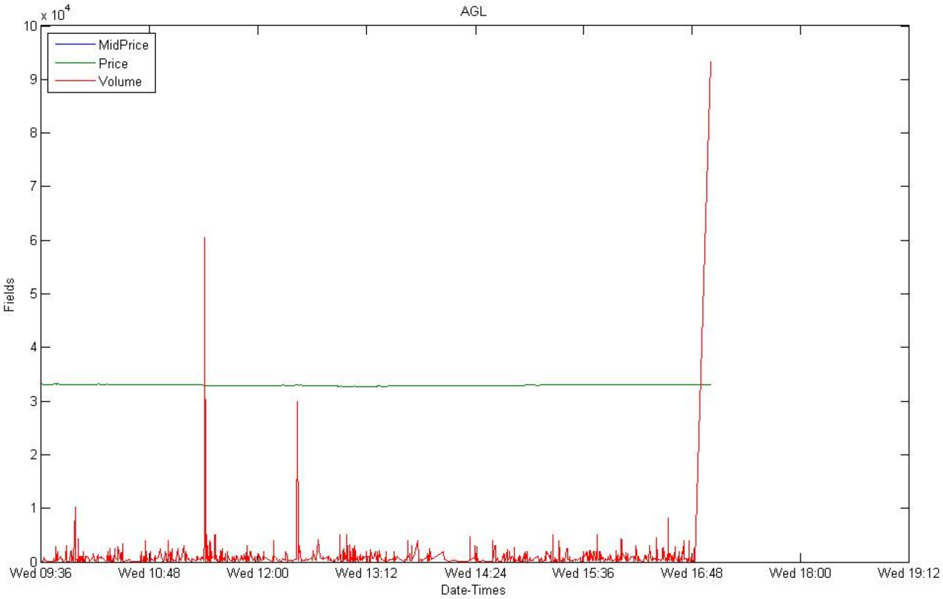
Class constructor and methods for Bar-Data (already aggregated)

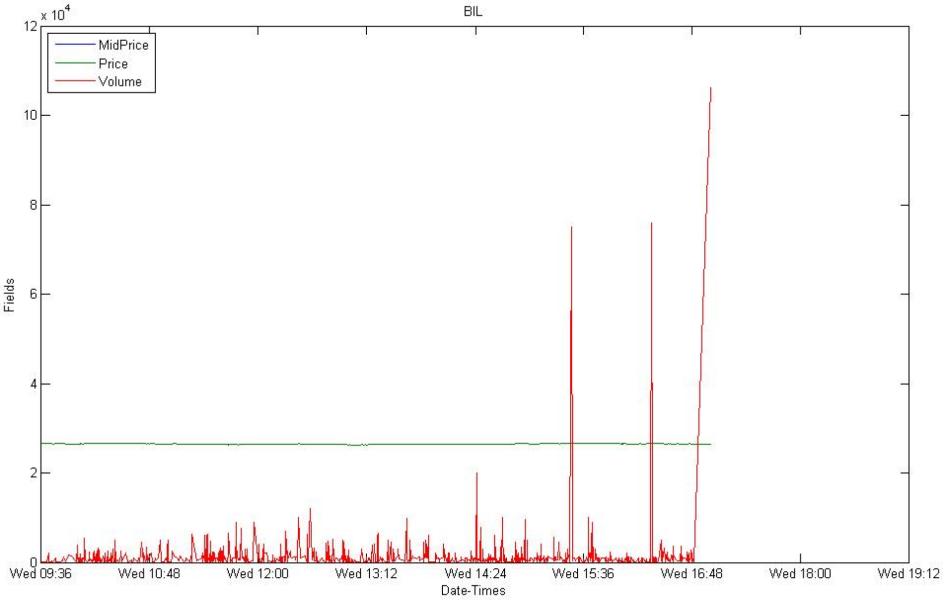
```
tsE = hfts(dataE),
```

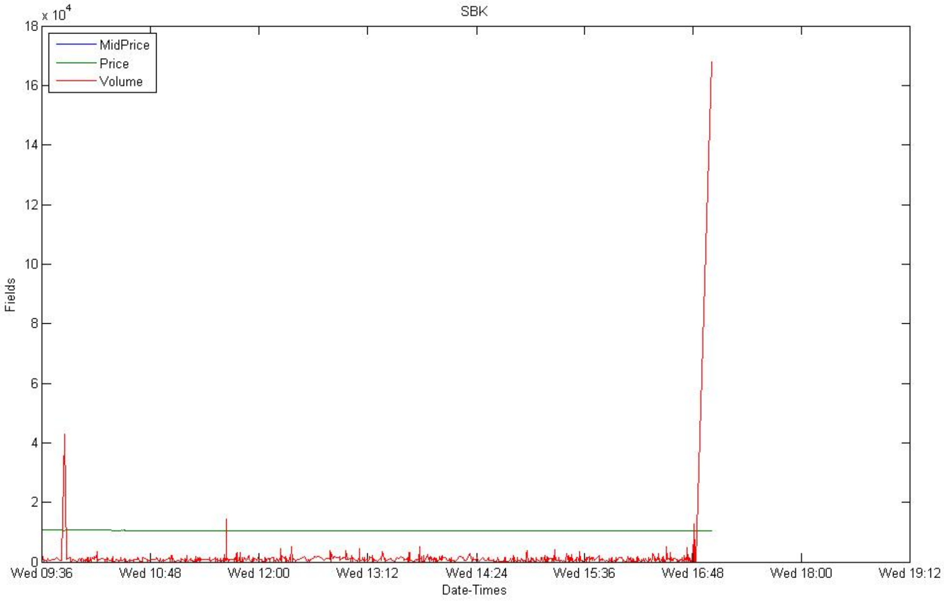
```
Warning: The class file for 'hfts' has been changed; but the change cannot
applied because objects based on the old class file still exist. If you us
those objects, you might get unexpected results. You can use the 'clear' c
to remove those objects. See 'help clear' for information on how to remove
objects.
  hfts

  Properties:
    freq: 'unknown'


 series : AGL, BIL
 fields : DateTime, CloseAsk, CloseBid, High, Last, Low, No_Asks, No_Bids,
```

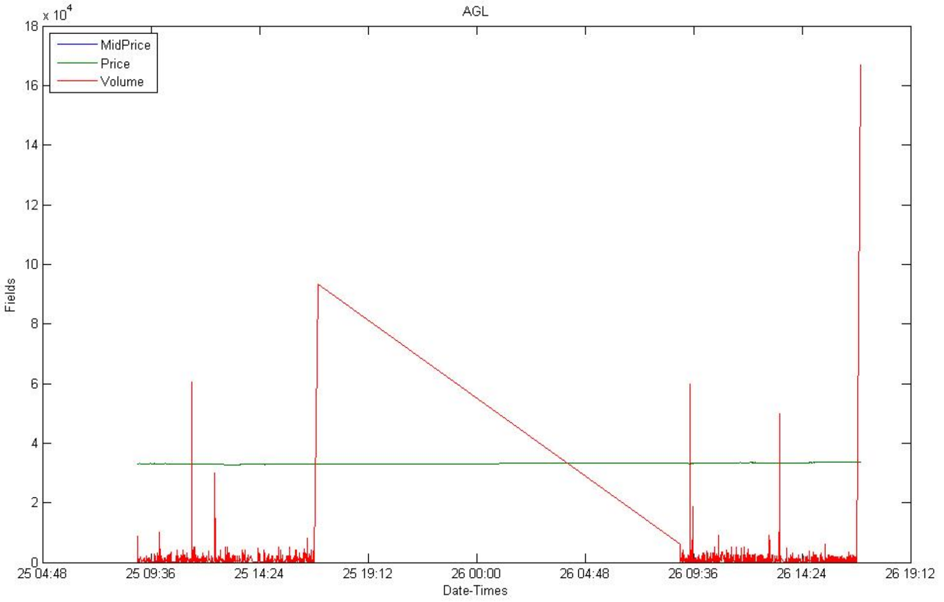
Plot function

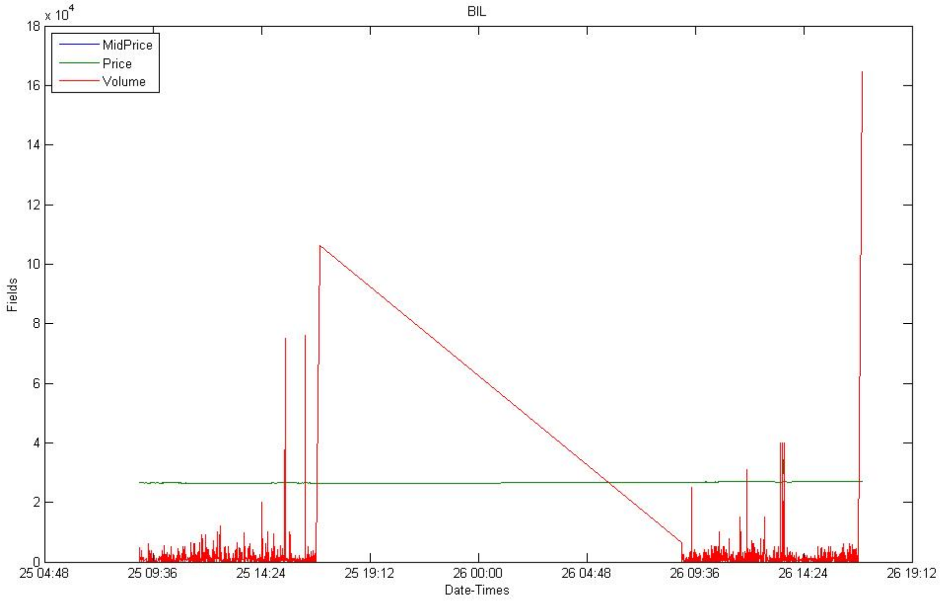
```
plot(tsH,'25-Jan-2012',1.5);
plot(tsH,'25-Jan-2012',10);
```

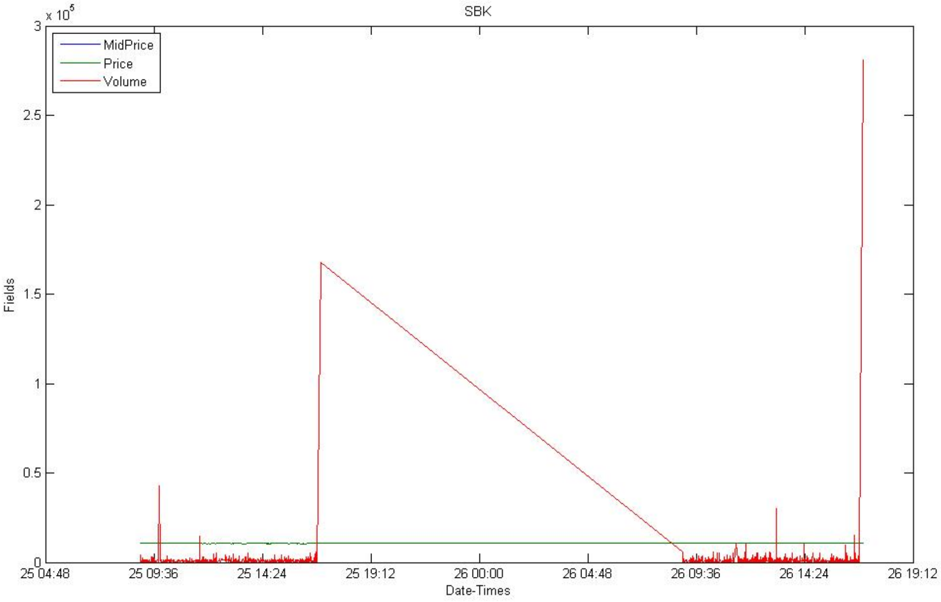

AGL
x 10^4
10
9
8
7
6
5
4
3
2
1
0
Fields
MidPrice
Price
Volume
Wed 09:36
Wed 10:48
Wed 12:00
Wed 13:12
Wed 14:24
Wed 15:36
Wed 16:48
Wed 18:00
Wed 19:12
Date-Times


BIL
x 10^4
12
10
8
6
4
2
0
Fields
MidPrice
Price
Volume
Wed 09:36
Wed 10:48
Wed 12:00
Wed 13:12
Wed 14:24
Wed 15:36
Wed 16:48
Wed 18:00
Wed 19:12
Date-Times


SBK
x 10^4
18
16
14
12
10
8
6
4
2
0
Fields
MidPrice
Price
Volume
Wed 09:36
Wed 10:48
Wed 12:00
Wed 13:12
Wed 14:24
Wed 15:36
Wed 16:48
Wed 18:00
Wed 19:12
Date-Times


AGL
x 10^4
18
16
14
12
10
8
6
4
2
0
Fields
MidPrice
Price
Volume
25 04:48
25 09:36
25 14:24
25 19:12
26 00:00
26 04:48
26 09:36
26 14:24
26 19:12
Date-Times





fill in missing data

linear interpolation

```
tsHf=fillts(tsH);
% zero-order hold
tsHfz=fillts(tsH,'z');
```

remove overlapping missing data

```
stH1r=nanfreets(tsH1);
display(stH1r);
```

```
Warning: All rows are NaN valued for MidPrice
  hfts

  Properties:
    freq: 'unknown'


 series : MidPrice, Price, RIC, Volume
 fields : DateTime, AGL, BIL, SBK
```

compute returns

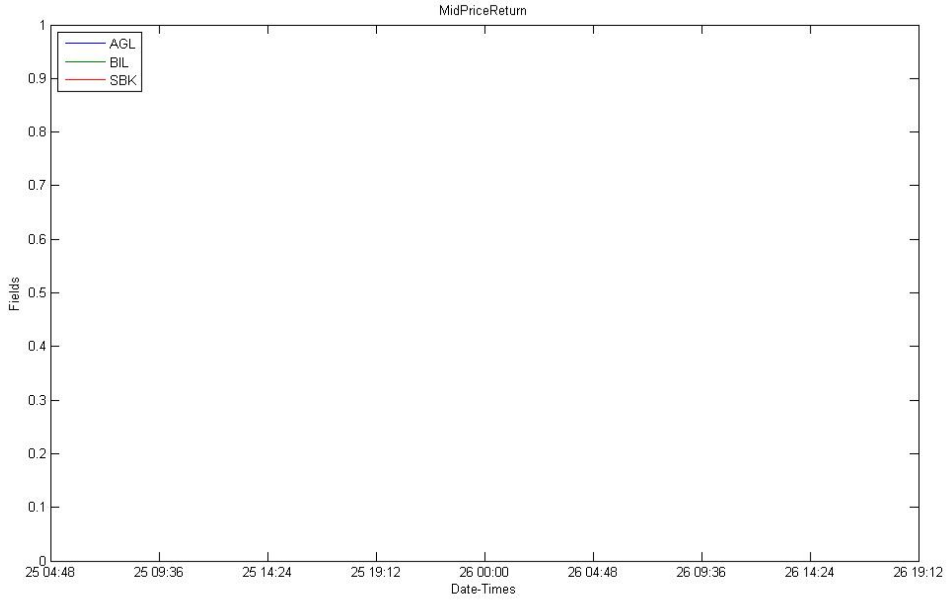
covert to tick-to-tick returns

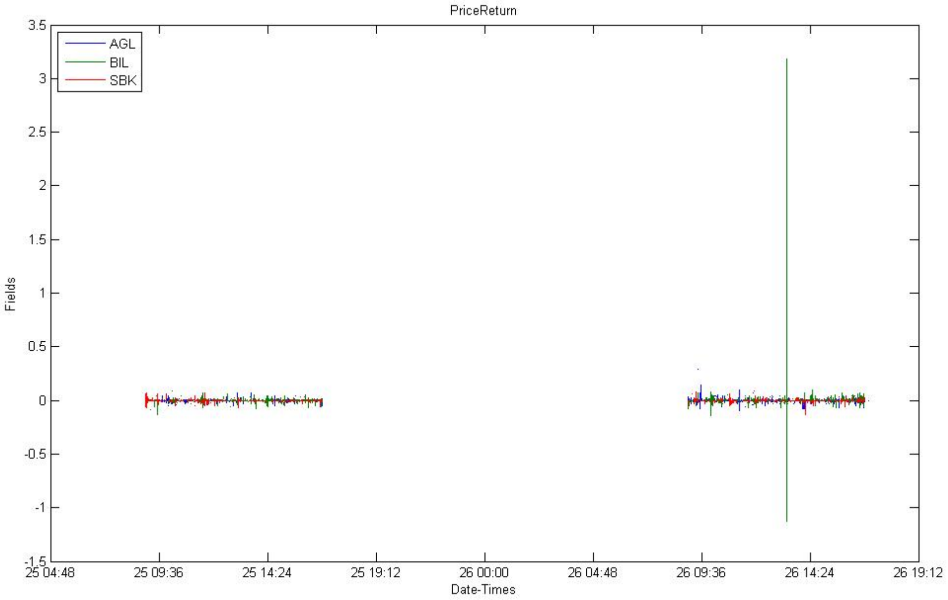
```
stH1ret = tick2ret(tsH1);
% convert to inhomogenously sampled per-minutes returns
stH1rets = tick2ret(tsH1,'geometric','ticktime');
display(stH1rets);
```

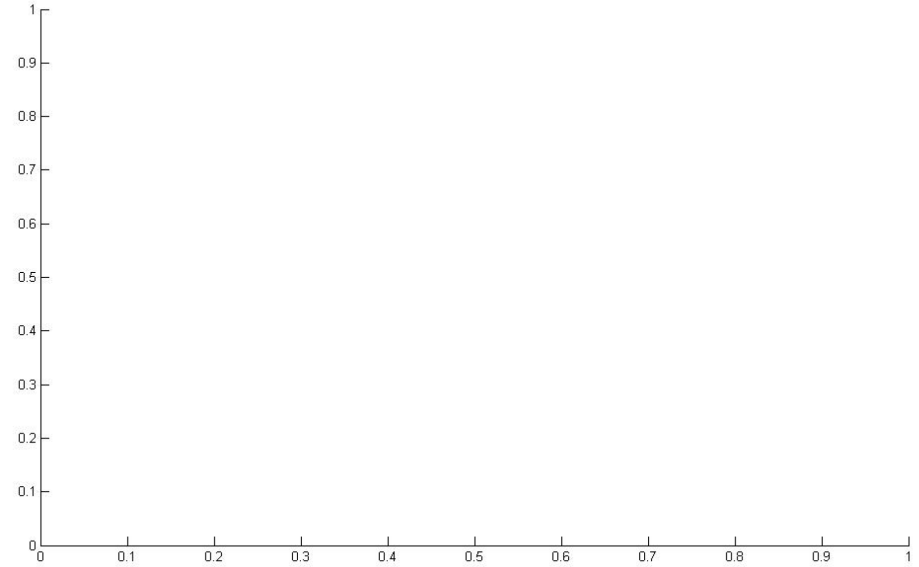
```
Warning: Untested
Warning: Untested
  hfts

  Properties:
    freq: 'unknown'


 series : MidPriceReturn, PriceReturn, RICReturn, VolumeReturn
 fields : DateTime, AGL, BIL, SBK
```

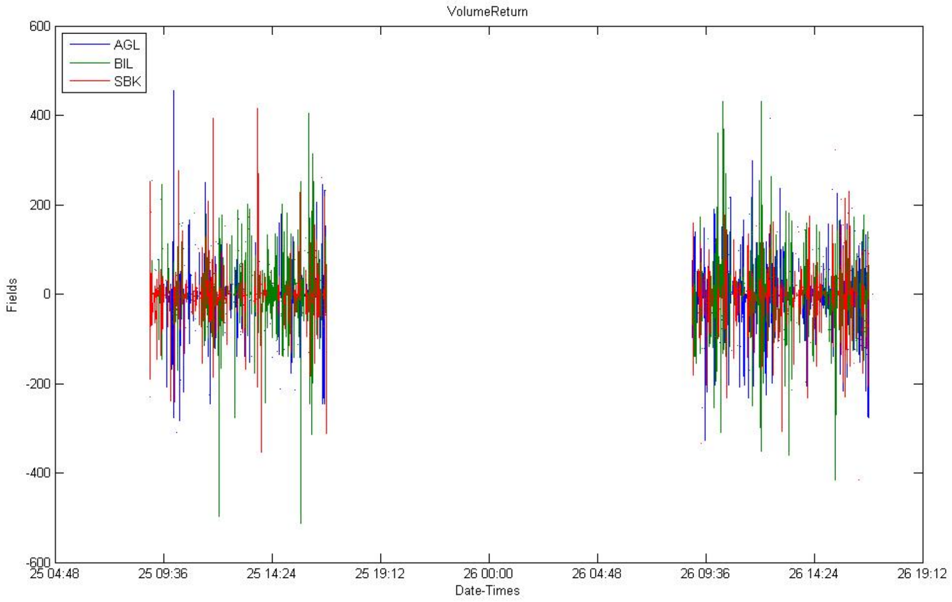
plot the price fluctuations

```
plot(stH1rets,'25-Jan-2012',10);
```


MidPriceReturn
AGL
BIL
SBK
Fields
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
25 04:48
25 09:36
25 14:24
25 19:12
26 00:00
26 04:48
26 09:36
26 14:24
26 19:12
Date-Times


PriceReturn
AGL
BIL
SBK
Fields
3.5
3
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
25 04:48
25 09:36
25 14:24
25 19:12
26 00:00
26 04:48
26 09:36
26 14:24
26 19:12
Date-Times





Documentation

```
help hfts
help hfts/aggregate
help hfts/extend
help hfts/fints
help hfts/mergets
help hfts/plot
help hfts/subsasgn
help hfts/tick2ret
help hfts/display
help hfts/fillts
help hfts/hfts
help hfts/nanfreets
help hfts/resample
help hfts/subsref
```

```
Warning: The class file for 'hfts' has been changed; but the change cannot
applied because objects based on the old class file still exist. If you us
those objects, you might get unexpected results. You can use the 'clear' c
to remove those objects. See 'help clear' for information on how to remove
objects.
Warning: The class file for 'hfts' has been changed; but the change cannot
applied because objects based on the old class file still exist. If you us
those objects, you might get unexpected results. You can use the 'clear' c
to remove those objects. See 'help clear' for information on how to remove
objects.
Contents of hfts:

hfts                           - class definition for High-Frequency Time-


hfts is both a directory and a function.

  HFTS class definition for High-Frequency Time-Series objects

  Allows both inhomogeneously sampled data as well as homogeneously
  sampled data as set by the frequency property FREQ. For inhomogenous
  data the frequency property is set to 'Unknown', this is the default
  setting. The class allows the frequency to take on the value of
  seconds or and minutes. The class allows resampling to uniform
  spacing that is not seconds or minutes and the frequency property
  takes on the value 'uniform'. The HFTS class allows frequencies
  lower than this; it is recommed that FINTS objects be used for
  frequencies lower than 1 minute. The class has a typecast method
  to FINTS objects. This type cast method will down-sampled the HFTS
  object to minutes and convert to a FINTS object.

  Table 1. Allowed FREQ values
  +-------------+-----------------------+---------------------------+
  | Frequency   | FREQ values           | Recommended Data class    |
  +-------------+-----------------------+---------------------------+
  | uniform     | 'uniform'             | HFTS (<1min) FINTS(>1min) |
  | 1-second    | 's','Sec'             | HFTS                      |
  | 1-minute    | 'm','Min','1m','1Min' | FINTS                     |
  | 10-minute   | '10m','10Min'         | FINTS                     |
  | 30-minute   | '30m','30Min'         | FINTS                     |
  | 1-hour      | '1h','1Hour'          | FINTS                     |
  | daily       | 'D','Daily'           | FINTS                     |
  | weekly      | 'W','Weekly'          | FINTS                     |
  | monthly     | 'M','Monthly'         | FINTS                     |
  | quarterly   | 'Q','Quarterly'       | FINTS                     |
  | semi-annual | 'S','Semi-annual'     | FINTS                     |
  | annual      | 'A','annual'          | FINTS                     |
  | unknown      | 'U','unknown'         | HFTS(<1min),FINTS(>=1min) |
  +-------------+-----------------------+---------------------------+

  The object is constructed from ENTITIES and ITEMS. The object is
  constructed so that each ENTITIES has uniquely enumerated ITEMS that
  can be inhomogeneously sampled. Intitial ENTITIES are mapped into
  SERIES in the object and the FIELDS of each SERIES are set to be the
  ITEMS. When an object is merged, it is merge on the DATETIME field
  over the ENTITIES. This then sets the ITEMS to be the SERIES and each
  SERIES of a given ITEM with have the ENTITY names as the FIELDS. Each
  SERIES is represented by a dataset object.

  HFTS object aggregates datatset objects for SERIES with fields FIELD
  I: unmerged HFTS : ENTITY,ITEM --> dataset of SERIES with fields FIELD
  II: merged HFTS  : ENTITY,ITEM --> dataset of FIELD with fields SERIES

  1. Time-series Data Aggregation:
    1.1. Raw HFT data is loaded either from the FDS, TRTH or a .CSV file.
    1.2. The constructor converts each ticker to a DATASET object
    1.3. AGGREGATE data to remove repeated simultaneous trades

  2. Time-series Merging
    2.1 MERGE time-series into a single dataset object per ITEM such as
    'Price' and 'Volume'

  3. Time-series Downsampling (resampling)
    3.1. RESAMPLE time-series objects to a uniform resampling frequency.
     After resampling the object is re-aggregated. Resampling is based
     on creating duplicate date-times in the object and the using the
     aggregation rules of AGGREGATE.
    3.2. RESAMPLE time-series objects to seconds and minutes.
    3.3. Type-cast time-series to a FINTS objects when the  minimum
     sampling frequency is 1-minute. When type-cast a structure is
     returned with a FINTS object for each SERIES in the object.

  Note 1: The HFTS class aggregates DATASET objects for each unique
    tickers when it is not a merged time-series. It aggregates DATASET
    objects for each unique fields when it is an merged time-series.
  Note 2: HFTS constructor expects fields 'RIC','DateL', and 'TimeL'.
  Note 3: AGGREGATE expects fields 'Volume' and 'Price'
  Note 4: This was based on the Chopper tools provided by Mathworks.

  See Also: FINTS, DATASET, HFTS/HFTS, MERGETS, AGGREGATE

Warning: The class file for 'hfts' has been changed; but the change cannot
applied because objects based on the old class file still exist. If you us
those objects, you might get unexpected results. You can use the 'clear' c
to remove those objects. See 'help clear' for information on how to remove
objects.
Warning: The class file for 'hfts' has been changed; but the change cannot
applied because objects based on the old class file still exist. If you us
those objects, you might get unexpected results. You can use the 'clear' c
to remove those objects. See 'help clear' for information on how to remove
objects.
```

```
  AGGREGATE Remove duplicate time information

  TS = AGGREGATE(TS) TS is an un-merged HFTS object

  Combines the price and volume information occurring at the same
  time in a time series. For every unique timestamp, trades occurring
  with that time stamp are aggregated so that the resulting volume
  is the sum of the volumes for each trade and the resulting price
  is a volume-weighted average of the prices at which each trade occurred.
  Aggregate before Merging times-series. Resampling is based on
  aggregation by duplicating time-stamps.

  See Also: HFTS/HFTS, MERGETS, EXTEND, RESAMPLE

Warning: The class file for 'hfts' has been changed; but the change cannot
applied because objects based on the old class file still exist. If you us
those objects, you might get unexpected results. You can use the 'clear' c
to remove those objects. See 'help clear' for information on how to remove
objects.
Warning: The class file for 'hfts' has been changed; but the change cannot
applied because objects based on the old class file still exist. If you us
those objects, you might get unexpected results. You can use the 'clear' c
to remove those objects. See 'help clear' for information on how to remove
objects.
  EXTENDTS fills in the missing times in a HF time-series

  See Also: HFTS

Warning: The class file for 'hfts' has been changed; but the change cannot
applied because objects based on the old class file still exist. If you us
those objects, you might get unexpected results. You can use the 'clear' c
to remove those objects. See 'help clear' for information on how to remove
objects.
Warning: The class file for 'hfts' has been changed; but the change cannot
applied because objects based on the old class file still exist. If you us
those objects, you might get unexpected results. You can use the 'clear' c
to remove those objects. See 'help clear' for information on how to remove
objects.
  FINTS Convert to FINTS object for intervals greater than 1-minute

  FTS = FINTS(TS) For TS of class HFTS. This will downsample to
    minutes. It will not correctly aggregate the time-series.
    To correctly aggregate first resample to 1-minute by
    setting the FREQ property to 'minute' or RESAMPLE to
    1/60 of an hour.

  See Also: HFTS/HFTS

Warning: The class file for 'hfts' has been changed; but the change cannot
applied because objects based on the old class file still exist. If you us
those objects, you might get unexpected results. You can use the 'clear' c
to remove those objects. See 'help clear' for information on how to remove
objects.
Warning: The class file for 'hfts' has been changed; but the change cannot
```

```
applied because objects based on the old class file still exist. If you us
those objects, you might get unexpected results. You can use the 'clear' c
to remove those objects. See 'help clear' for information on how to remove
objects.
  MERGETS Merges all the HFT time-series in object on Time

  TS = MERGETS(TS) Combines all the HFT time-series onto the
    same Date and Time range. The resulting time series has
    timestamps that are the union of the timestamps of the
    two original series. Any missing prices are represented
    as NaN.

  Note: To merge two HFTS objects see HFTS/MERGE

  See Also: HFTS/HFTS, AGGREGATE, RESAMPLE, MERGE

Warning: The class file for 'hfts' has been changed; but the change cannot
applied because objects based on the old class file still exist. If you us
those objects, you might get unexpected results. You can use the 'clear' c
to remove those objects. See 'help clear' for information on how to remove
objects.
Warning: The class file for 'hfts' has been changed; but the change cannot
applied because objects based on the old class file still exist. If you us
those objects, you might get unexpected results. You can use the 'clear' c
to remove those objects. See 'help clear' for information on how to remove
objects.
  PLOT plots sections of a time series.

  H = PLOT(TS,DATETIMES,WINDOW) WINDOW is in fractional
    units of days, DATETIME are date-times and TS is an HFTS
    object.

  It accepts a vector of dates that form the center of each
  plot and a window of time duration around those dates.

  See Also

Warning: The class file for 'hfts' has been changed; but the change cannot
applied because objects based on the old class file still exist. If you us
those objects, you might get unexpected results. You can use the 'clear' c
to remove those objects. See 'help clear' for information on how to remove
objects.
Warning: The class file for 'hfts' has been changed; but the change cannot
applied because objects based on the old class file still exist. If you us
those objects, you might get unexpected results. You can use the 'clear' c
to remove those objects. See 'help clear' for information on how to remove
objects.
  SUBSASGN Assign HFTS object properties

  FREQ can be assigned 's','m', or 'u'

Warning: The class file for 'hfts' has been changed; but the change cannot
applied because objects based on the old class file still exist. If you us
those objects, you might get unexpected results. You can use the 'clear' c
```

```
to remove those objects. See 'help clear' for information on how to remove
objects.
Warning: The class file for 'hfts' has been changed; but the change cannot
applied because objects based on the old class file still exist. If you us
those objects, you might get unexpected results. You can use the 'clear' c
to remove those objects. See 'help clear' for information on how to remove
objects.
  TICK2RET Compute returns for HFTS object

  TS = TICK2RET(TS)

  TS = TICK2RET(TS,TYPE)

  TS = TICK2RET(TS,TYPE,SCALING)

  Table 1: Return Types
  +---------------+--------------------+------------------+
  | TYPE          | METHOD             | DESCRIPTION      |
  +---------------+--------------------+------------------+
  |'Geometric'**  | DIFF(LN(P))        | LN(P(T)/P(T-1))  |
  |'PriceRelative'| EXP(DIFF(LN(P))    | P(T)/P(T-1)      |
  |'Arithmetic'   | EXP(DIFF(LN(P))-1) | (P(T)/P(T-1))-1  |
  +---------------+--------------------+------------------+
  Table 1: Scaling Types
  +---------------+-----------------------------------+
  | SCALING       | DESCRIPTION                       |
  +---------------+-----------------------------------+
  |'TickTime'*    | Rescaled by the time-change (tau)  |
  |               | between ticks. This homogenises    |
  |               | returns in terms of the rate of    |
  |               | trading.[Per Minutes (Ret/(24*60)] |
  |'DataTime'**   | No rescaling. For uniform returns  |
  |               | resample the data to Bar data      |
  |               | first.                             |
  +---------------+-----------------------------------+
  * P = P_0 EXP(RT) <=> P = P_0 EXP((R_0/TAU) T)
  ** Default value

  See Also: NANFREETS, RESAMPLE, FILLTS

Warning: The class file for 'hfts' has been changed; but the change cannot
applied because objects based on the old class file still exist. If you us
those objects, you might get unexpected results. You can use the 'clear' c
to remove those objects. See 'help clear' for information on how to remove
objects.
Warning: The class file for 'hfts' has been changed; but the change cannot
applied because objects based on the old class file still exist. If you us
those objects, you might get unexpected results. You can use the 'clear' c
to remove those objects. See 'help clear' for information on how to remove
objects.
  DISPLAY Display a High-Frequency Time-series object

  See Also: DISP,
Warning: The class file for 'hfts' has been changed; but the change cannot
applied because objects based on the old class file still exist. If you us
those objects, you might get unexpected results. You can use the 'clear' c
to remove those objects. See 'help clear' for information on how to remove
objects.
Warning: The class file for 'hfts' has been changed; but the change cannot
applied because objects based on the old class file still exist. If you us
those objects, you might get unexpected results. You can use the 'clear' c
to remove those objects. See 'help clear' for information on how to remove
objects.
  FILLTS Fill in missing data in HFTS object

  TS = FILLTS(TS) use INTERP1 fill types

  TS = FILLTS(TS,TYPE) us fill type TYPE.

  See Also: INTERP1, ZEROORDERHOLD

Warning: The class file for 'hfts' has been changed; but the change cannot
applied because objects based on the old class file still exist. If you us
those objects, you might get unexpected results. You can use the 'clear' c
to remove those objects. See 'help clear' for information on how to remove
objects.
Warning: The class file for 'hfts' has been changed; but the change cannot
applied because objects based on the old class file still exist. If you us
those objects, you might get unexpected results. You can use the 'clear' c
to remove those objects. See 'help clear' for information on how to remove
objects.
  hfts Constructor for hfts class

  HFTS = HFTS(DATA) DATA is a cell-array in relational format
    in the order of TICKERS,LOCALDATE,LOCALTIME,PRICE,VOLUME.
    Expects the data to have a field for each STOCK where the
    fieldname is the STOCK ticker. The fieldname does not have
    to be the same as the populated ticker name in the data
    cell-array.

  HFTS = HFTS(DATA,ITEM) Only keep items in ITEMS. If ITEM is
    not set all avaliable unique ITEMS in DATA will be used.
    The order of ITEMS is preserved. All ITEMS should reflect
    in all the DATA sets.

  Example 1: Recommended constructor for 'Trade' data
  >> ts = hfts(dataG,{'Price','Volume'}),
  hfts

   Properties:
    freq: 'unknown'

   Methods

  series : AGL, BIL
  fields : RIC, DateTime, Price, Volume
```

```
Example 2: Default construction
>> ts
hfts

Properties:
  freq: 'unknown'

Methods

series : AGL, BIL
fields : DateTime, MidPrice, Price, RIC, Type, Volume

Note 1: The DATA structure can be prepared in two distinct
  ways. First, directly from the FDS, Second, using TRTH.

Method 1: From FDS using a valid database connection conn
1.1. Bar Data
  >> data = fetchtrth(conn,{'AGLJ.J'},'Intraday 10Min',[datenum('2-Oct-2

  data.AGL(1:5,:) =

 Columns 1 through 10
   'RIC'       'DateL'         'TimeL'       'Type'              'Open'
   'AGLJ.J'    '2011-10-03'    '09:00:00'    'Intraday 10Min'    [27200]
   'AGLJ.J'    '2011-10-03'    '09:10:00'    'Intraday 10Min'    [27300]
   'AGLJ.J'    '2011-10-03'    '09:20:00'    'Intraday 10Min'    [27401]
   'AGLJ.J'    '2011-10-03'    '09:30:00'    'Intraday 10Min'    [27370]

 Columns 11 through 17
   'CloseBid'    'No_Bids'    'OpenAsk'    'CloseAsk'    'No_Asks'    'H
   [   27300]    [   2388]    [  26999]    [   27349]    [   2388]    [
   [   27383]    [   1046]    [  27349]    [   27445]    [   1046]    [
   [   27333]    [   1033]    [  27445]    [   27370]    [   1033]    [
   [   27290]    [   2151]    [  27370]    [   27325]    [   2151]    [

1.2. Trade Data
  >> data = fetchtrth(conn,{'AGLJ.J','BILJ.J'},'Trade',[datenum('2-Oct-2

      AGL: {20278x7 cell}
      BIL: {18327x7 cell}

  >> data.AGL =

  'RIC'       'DateL'         'TimeL'       'Type'     'Price'    'Volum
  'AGLJ.J'    '2011-12-15'    '09:00:28'    'Trade'    [29650]    [  454
  'AGLJ.J'    '2011-12-15'    '09:00:31'    'Trade'    [29640]    [   30
  'AGLJ.J'    '2011-12-15'    '09:01:22'    'Trade'    [29650]    [
  'AGLJ.J'    '2011-12-15'    '09:01:31'    'Trade'    [29650]    [   37

Method 2: Using TRTH using a valid TRTH connection r
2.1. TRTH data
  >> data = trth2struct(r,'AGLJ.J',{'Price','Volume','Mid Price'},'2-Jan

Method 3: Using RMDS created real-time data
```

```
  3.1. RMDS data
    >> rconn = reuters(rc.session,rc.source,rc.id,[],1);
    >> [data,ts0] = fetchreuters(rconn,{'AGLJ.J','BILJ.J'},{'TRDPRC_1'});
    >> data = reuters2cell(ts0,{'TRDPRC_1','TRDVOL_1'},{'Price','Volume'})
    >> data = hfts(data).
   HFTS
    Properties:
    freq: 'unknown'
   METHODS

    series : AGL, BIL
    fields : DateTime, Price, RIC, Volume
    >> data.AGL
    ans =
        DateTime      Price    RIC      Volume
        7.3497e+05    28352    AGLJ.J    45

  See Also: HFTS/FINTS, HFTS.FREQ

Warning: The class file for 'hfts' has been changed; but the change cannot
applied because objects based on the old class file still exist. If you us
those objects, you might get unexpected results. You can use the 'clear' c
to remove those objects. See 'help clear' for information on how to remove
objects.
Warning: The class file for 'hfts' has been changed; but the change cannot
applied because objects based on the old class file still exist. If you us
those objects, you might get unexpected results. You can use the 'clear' c
to remove those objects. See 'help clear' for information on how to remove
objects.
  NANFREETS Remove rows with missing data in each series

  See Also: HFTS/FILLTS

Warning: The class file for 'hfts' has been changed; but the change cannot
applied because objects based on the old class file still exist. If you us
those objects, you might get unexpected results. You can use the 'clear' c
to remove those objects. See 'help clear' for information on how to remove
objects.
Warning: The class file for 'hfts' has been changed; but the change cannot
applied because objects based on the old class file still exist. If you us
those objects, you might get unexpected results. You can use the 'clear' c
to remove those objects. See 'help clear' for information on how to remove
objects.
  RESAMPLE Downsample the HFTS object

  TRD = RESAMPLE(TS,SPACING)

  Reduces the time resolution of a time series, thereby
  increasing the spacing between ticks. The spacing must be
  specified in fractions of hours. Use this method with
  AGGREGATE to aggregate all the resulting ticks that have
  the same timestamp.

  Note 1: The prefered method is to set the frequency property.
  See Also: HFTS/HFTS, AGGREGATE, MERGETS

Warning: The class file for 'hfts' has been changed; but the change cannot
applied because objects based on the old class file still exist. If you us
those objects, you might get unexpected results. You can use the 'clear' c
to remove those objects. See 'help clear' for information on how to remove
objects.
Warning: The class file for 'hfts' has been changed; but the change cannot
applied because objects based on the old class file still exist. If you us
those objects, you might get unexpected results. You can use the 'clear' c
to remove those objects. See 'help clear' for information on how to remove
objects.
  HFTS.SUBSREF Subscript reference HFTS object

  The properties FREQ and SERIES can be subscript referenced.
  The SERIES property is a dynamics property based on the state
  of the object. If the object has been merged then the series
  are the ITEMS in the orginal data, e.g. Price and Volume:
  >> ts
  hfts

   Properties:
     freq: 'unknown'

   Methods

  series : Price, Volume
  fields : DateTime, AGL, BIL

  >> ts.Price(1:2,:)
  ans =
    DateTime      AGL    BIL
    7.3485e+05    NaN    23949
    7.3485e+05    NaN    23902

  >> ts.Volume(1:2,:)
  ans =
    DateTime      AGL    BIL
    7.3485e+05    0      4778
    7.3485e+05    0        50

  If the object has not been merged then SERIES are the ticker
  names of the ENTITIES in the original data e.g.
  >> ts
  hfts

  Properties:
    freq: 'unknown'

  Methods

    series : AGL, BIL
    fields : RIC, DateTime, Price, Volume
```

```
>> ts.AGL(1:2,:)
ans =
 RIC       DateTime      Price    Volume
  AGLJ.J    7.3485e+05    29650    4542
  AGLJ.J    7.3485e+05    29640     300

>> ts.AGL.DateTime(1:2,:)
  ans =
      1.0e+05 *
          7.3485
          7.3485

See Also:
```

Published with MATLAB® 7.14