Data clean up

Shiyuan Wang

5/25/2021

```
###LIBRARIES
library(lubridate)
library(highfrequency)
library(stats)
library(xts)
```

```
trade_data <- read.csv("trade_file.csv")
trade_data$X <- NULL
colnames(trade_data) <- c("timestamp", "type", "exchange_code", "symbol","price","Size")
head(trade_data$timestamp)
```

```
## [1] "2020-11-23 09:30:12.643838+00:00" "2020-11-23 09:30:12.643838+00:00"
## [3] "2020-11-23 09:30:12.768285+00:00" "2020-11-23 09:30:14.340402+00:00"
## [5] "2020-11-23 09:30:26.095709+00:00" "2020-11-23 09:30:32.723609+00:00"
```

```
###Format the the timestamp
trade_data$timestamp <- as.character(trade_data$timestamp)
trade_data$timestamp <- substr(trade_data$timestamp, 1, 26)

my_options <- options(digits.secs = 6)
trade_data$timestamp <- strptime(trade_data$timestamp, "%Y-%m-%d %H:%M:%OS", tz = "EST")

###summary of data
summary(trade_data[, c("timestamp","price","Size")])
```

```
##    timestamp                         price            Size
##  Min.   :2020-11-23 09:30:12.64   Min.   :115.1   Min.   :      1.0
##  1st Qu.:2020-11-23 14:46:51.36   1st Qu.:115.5   1st Qu.:      5.0
##  Median :2020-11-23 15:10:05.20   Median :116.1   Median :     50.0
##  Mean   :2020-11-23 15:06:08.66   Mean   :116.3   Mean   :    115.8
##  3rd Qu.:2020-11-23 15:28:54.02   3rd Qu.:117.1   3rd Qu.:    100.0
##  Max.   :2020-11-23 15:59:59.99   Max.   :117.8   Max.   :1143860.0
##                                   NA's   :1       NA's   :1
```

```
a <- xts(trade_data,order.by=as.POSIXct(trade_data$timestamp))
trades_afterclean <- tradesCleanup(tDataRaw= a,exchanges = "NSQ", tz = "EST")
```

```
quote_data <- read.csv("quote_file.csv")
quote_data$X <- NULL
```

```
colnames(quote_data) <- c("DT2", "type", "EX", "symbol","BID","BIDSIZ","OFR","OFRSIZ")
quote_data <- quote_data[,c("DT2","EX","BID","BIDSIZ","OFR","OFRSIZ","symbol")]
quote_data$exchange_code <- "T"

###Format the the timestamp
quote_data$DT2 <- as.character(quote_data$DT2)
quote_data$DT2 <- substr(quote_data$DT2, 1, 26)

my_options <- options(digits.secs = 6)

quote_data$DT2 <- strptime(quote_data$DT2, "%Y-%m-%d %H:%M:%OS", tz = "EST")


b <- as.xts(quote_data,order.by=as.POSIXct(quote_data$DT2))

quotes_afterclean <- quotesCleanup(qDataRaw= b)
```

```
## [1] "The   is the exchange with the highest volume."
```

```
#DATA Clean up function

aggregatePrice()
###Aggregate a times series but keep first and last observations.

aggregateQuotes()
###Aggregate a quote data in a xts format

aggregateTrades()
###Aggregate a trade data in a xts format

aggregateTS()
###Aggregate a time series, it did pretty much the same thing as the aggregatePrice.

tradesCleanup()
###This function is a wrapper function for cleaning the trade data.
###It must contain columns: DT2, exchange code, SYMBOL, PRICE, SIZE ,BID

quotesCleanup()
###This function is a wrapper function for cleaning the quote data.
###It must contain columns: DT2, SYMBOL, EX, BID, BIDSIZ, OFR, OFRSIZ, PRICE)
###For trades/quotes clean up function, it requires xts format, so if we have cvs file, it can automati

autoSelectExchangeQuotes()
###Only return the data from the stock exchange with the highest volume in quote data

autoSelectExchangeTrades()
###Only return the data from the stock exchange with the highest trading volume in trade data

businessTimeAggregation()
###Aggregation function based on business time.

exchangeHoursOnly()
###This function is used for extracting data from an xts object for the exchange hours only.
```

```
makeOHLCV()
###this function is a kind of aggregation function that can make the high frequency data become OHLCV d

makeRMFormat()
###this function is used for spliting data to a format which can be used for realized measure.

matchTradesQuotes()
###this function can match trade data and quote data and combine them.

mergeQuotesSameTimestamp()
mergeTradesSameTimestamp()
###this function is also aggregating the quote/trade data which has the same timestamp.
```

```
#Statistical test

AJjumpTest()
### Ait-Sahalia and Jacod test for the presence of jumps in the price series.

BNSjumpTest()
### Barndorff-Nielsen and Shephard tests for the presence of jumps in the price series.
###Null hypothesis: there are no jumps.

driftBursts()
###This function will return the result of testing drift burst hypothesis and also shows the test stati

# dat <- sampleTData[as.Date(DT) == "2018-01-02"]
# DBH <- driftBursts(dat, testTimes = seq(35100,57600,60), preAverage = 2, ACLag = -1L, meanBandwidth =
# print(DBH)

getCriticalValues()
###get critical values for drift burst hypothesis

getLiquidityMeasures()
###Compute Liquidity Measures

intradayJumpTest()
###This can be used to test for jumps in intraday price paths.

IVinference()
###This function returns the SE, value and confidence band of Integrated variance estimator.

JOjumpTest()
###Test for jumps in the price series by using Jiang and Oomen test.

makePsd()
###this function can return the positive semidefinite projection of a symmetric matrix using the eigenv

rankJumpTest()
###Calculate the rank jump test of Li et al.

rAVGCov()
###Calculates realized variance by averaging across partially overlapping grids.
```

```
rBPCov()
###Calculate the Realized BiPoweer Covariance defined by Barndorff-Nielsen and Shephard.
```

```
#Building model

getTradeDirection()
###Using Lee and Ready algorithm to determine the inferred trade direction.

HARmodel()
###This function returns the estimates for the HAR model for realized volatility.

HEAVYmodel()
###This functions calculate HEAVY model which is introduced by Shepard and Sheppard.
```

```
#general information

listAvailableKernels()
###This function will list all available kernels

listCholCovEstimators()
###This function will list the available estimators for the CholCov estimation
```

```
library(TAQMNGR)
```

```
## Warning: package 'TAQMNGR' was built under R version 4.0.5
```

```
## ------------------------------------------------------------------------
## --                              TAQMNGR                                 --
## ------------------------------------------------------------------------
##     Package attached
```

```
dirInput <- "D:/desktop/Vanguard_research/1"
dirOutput <- "D:/desktop/Vanguard_research/2"

TAQ.CleanTickByTick(dirInput = dirInput, dirOutput = dirOutput,window = 80, deltaTrimmed = 0.10, granula
```

```
## The folder doesn't contain files to be cleaned.
```

```
## [1] 0
```

```
TAQ.Report(dirInput = dirOutput, symbol = c("DOG"))
```

```
## ##################################
## #     DAILY CLEANING REPORT      #
## ##################################
##
## Directory: D:\desktop\Vanguard_research\2
## Symbol: DOG
##
## DATE       #TRADES     NOTCORR_DELAY    BROWN_GALLO
```

```
## 20130701 5002    1   1
## 20130702 7859    15  5
## 20130703 6742    9   8
## 20130706 3690    4   1
## 20130707 3620    6   5
## 20130708 2823    4   5
## 20130709 2503    4   0
## 20130710 4554    3   5
## 20130713 4724    4   1
## 20130714 5615    2   4
## 20130715 9342    19  4
## 20130716 5830    7   4
## 20130717 3575    4   2
## 20130721 4757    10  12
## 20130722 3570    8   1
## 20130723 3615    1   0
## 20130724 3889    4   2
## 20130727 5857    12  3
## 20130728 4919    5   6
##
## ###################################
## #     TOTAL CLEANING REPORT     #
## ###################################
##
## Directory: D:\desktop\Vanguard_research\2
## Symbol: DOG
##
## #TRADES   NOTCORR_DELAY   BROWN_GALLO     NOTCORR_DELAY(%%)   BROWN_GALLO(%%)
## 92486    122 69  0.131912    0.0746059

## [1] 0
```

```
TAQ.Report(dirInput = dirOutput, symbol = c("GNU"))
```

```
## ###################################
## #     DAILY CLEANING REPORT     #
## ###################################
##
## Directory: D:\desktop\Vanguard_research\2
## Symbol: GNU
##
## DATE      #TRADES     NOTCORR_DELAY   BROWN_GALLO
## 20130701 5002    1   1
## 20130702 7859    15  9
## 20130703 6742    9   9
## 20130706 3690    4   1
## 20130707 3620    6   5
## 20130708 2823    4   5
## 20130709 2503    4   0
## 20130710 4554    3   5
## 20130713 4724    4   1
## 20130714 5615    2   4
## 20130715 9342    19  5
```

```
## 20130716 5830    7   4
## 20130717 3575    4   7
## 20130721 4757    10  13
## 20130722 3570    8   1
## 20130723 3615    1   0
## 20130724 3889    4   4
## 20130727 5857    12  4
## 20130728 4920    5   7
##
## ##################################
## #     TOTAL CLEANING REPORT      #
## ##################################
##
## Directory: D:\desktop\Vanguard_research\2
## Symbol: GNU
##
## #TRADES   NOTCORR_DELAY   BROWN_GALLO     NOTCORR_DELAY(%%)   BROWN_GALLO(%%)
## 92487    122 85  0.13191 0.0919048
```

```
## [1] 0
```

```
TAQ.Aggregate(dirInput = dirOutput, symbol = c("DOG", "GNU"), bin = 300, useAggregated = TRUE)
```

```
##
## Aggregating DOG data:
## The folder doesn't contain files to be aggregated.
##
## Aggregating GNU data:
## The folder doesn't contain files to be aggregated.
```

```
dog <- TAQ.Read(dirInput = dirOutput, symbol = "DOG", startDate = 00010101, endDate = 20141231, bin = 3
```