Vschon User Manual

1. Installation

1.1 prerequisite

(1) Python 2.7.6

(2) Python module: numpy

(3) Pyhton module: pandas

(4) Python module: sklearn

(5) Python module: scipy

(6) Python module: kdb

(7) Cython

(8) KDB

On 64 bit machine: TODO

1.2 VSHON package Installation

(1) Create VSCHON directory in /path/to/desired/VSCHON

(2) Copy the entire VA\_PYTHON directory to /path/to/VSCHON/VA\_PYTHON

(3) Copy the entire VD\_KDB directory to /path/to/VSHON/VD\_KDB

(4) Put following lines in .bashrc

export VSCHON=/path/to/VSCHON

export PYTHONPATH=$PYTHONPATH:$VSCHON/VA\_PYTHON

export PYTHONPATH=$PYTHONPATH:$VSCHON/VD\_KDB

1.3 VSCHON DATA Installation

DATA directory is the root directory of all databases. Currently, there are two complete databases in DATA:

(1) forex\_taqDB: stores the best bid/ask change of 15 major currency pairs from 2009 May to 2013 July.

(2) Edgar: stores the form10Q/K(Quarterly and annual report to SEC) of all US stocks from 1993 to 2013.

All the databases are managed by the "datamanage" submodule in VA\_PYTHON package. To properly use the database, please:

(1) Create Data directory in /path/to/Data

(2) Add following line to .bashrc:

export DATA=/path/to/Data

(3) Copy the forex\_taqDB and Edgar to DATA.

2. Overview of VSCHON package

There are three main components that consists VSCHON:

(1) VSCHON ANALYSIS(VA\_PYTHON, or va)

It is the major component for model analysis, trading stratrgy backtesting.Written in python and Cython.

(2) VSCHON DATABASE Management(VD\_KDB, or vd): This component contains the database manager for each database. It also contains the python API to communicate with KDB database.

(3) VSCHON DATA: All databases are physically stored in DATA.

3. VD\_KDB(vd)

VD\_KDB contains database manager for each database. Also, it has python API for KDB.

TODO: Restructure the module in following way

3.1 Module: vd.databaseManager

3.1.1 Class vd.databaseManager.forex\_taqManager

(1) forex\_taq Database

forex\_taq database stores the best bid and ask price of 15 major currency pairs. The data ranges from 2009 May to present.

(2) forex\_taq attributes

(3) forex\_taq methods

linkKDB(port=5000)

link to the KDB database

Params:

port(int): the port number of the Q process.

update()

download latest data and update the database

Output:

Updatelog.txt: record the success state and updated months.

summary():

summarize the state of forex\_taq databae, including symbol list, date range.

loadPrice(symbol, beginDate, endDate = beginDate):

load price series into memory

Params:

symbol(str): symbol to be loaded

beginDate(str): begin date of the price series, in format "2000.01.01"

EndDate(str): end daye of the price, default to be begin date

Q\_sql(directive):

Passing directive string to Q process and return corresponding data.

Params:

directive(str): the directive to be sent to Q process.

Return:

data(DataFrame): data to be returned

(4) Example

TODO: add examples

3.2 Submodule vd.KDBAPI

This submodule contains functions and classes for connecting to and retrieving data from. KDB.

3.2.1 Class dataLoader:

dataLoader is used to connect to KDB and retrieve data from it.

Similar to forex\_taqManager

4. VA\_PYTHON

4.1 Submodule models

This module contains models/algos used for financial data modelling.

4.1.1 Submodule hawkes

4.1.1.1 Class hawkes

It is an implementation of bivariate hawkes process to model high frequency tick data.

(1) hawkes attributes

(2) hawkes methods

(3) Examples

from va.models.hawkes import hawkes

myHawkes = hawkes()

myHawkes.fit(Price)

pred = myHawkes.predict(ahead = 2)

4.2 Submodule simulator

This module contains the simulator for backtesting trading strategies.