TASK 3 - Feedhandler Concepts

TASK 2 - Function Projections

TASK 4 - Parsing Messages (Part 1)

NECESSARY KNOWLEDGE

- n/a

I. AIM

The aim of this task is to understand the protocol used by our data source simulator (*simu.q*). To keep things simple, the protocol is fairly straightforward.

II. INFORMATION

Messages arrive into the feedhandler (**FH**) in key/value pairs, and they arrive nested.

Example:

```
("1=BARC.L;4=285.1;5=285.25;6=5172;7=5735";"1=VOD.L;4=196.3;5=196.4;6=14700;7=68414";"1=
```

The indiviual messages look like:

```
trade msg: "1=VOD.L;3=200.014;2=631044"
quote msg: "1=VOD.L;4=195;5=200;6=40000;7=5258"
```

The purpose of the feedhandler is to take this message and parse it to a format appropriate for the tickerplant process (**TP**). In other words, we need to transform string data into q objects/data types.

Once we have separated the nested messages we can break down the individual messages further. Looking at the "trade msg:" above we can see three pieces of information which are delimited by the semi-colon character.

```
1=VOD.L
3=200.014
2=631044
```

Each of the key values correspond to a particular field identifier (**FID**). These **FID**s are listed below.

```
Message FIDs:

1 - sym
2 - size
3 - price
4 - bid
5 - ask
6 - bsize
7 - asize
```

If there is only one message, assume it is enlisted, for example:

```
enlist "1=VOD.L;2=100.10;3=500"
```

NOTE:

The order of **FID**-value pairs is not guaranteed i.e.

```
"1=BASFn.DE;3=66.58;2=8"
```

as well as

```
"1=BASFn.DE;2=8;3=66.58"
```

are both valid messages.

III. VISUALISING MESSAGES

Open fh.q in your text editor and find the line which starts with

```
upd:{[msgs]
```

Add the line:

```
upd:{[msgs] 0N! msgs};
```

after the current upd function definition. The 0N! simply prints it's argument. Save and exit your text editor.

In one putty session start the feedhandler with the command:

```
q fh.q -tp localhost:5000 -p 4000
```

*NOTE: *If the script will not run and kdb complains the address is already being used, then try changing the port number. Port numbers below 1024 should not be used as these are reserved by linux.

To set a wider console view enter

```
\c 20 150
```

This will make your console 20 rows by 150 characters.

In a second putty session launch the simulator with:

```
q simu.q -fh localhost:4000 -data data/msgs
```

IV. TESTING

To test if everything is working as it should be return to the session which is running simu.q and enter:

```
do[20;pub[]]
```

This should publish 20 events. Return to the fh.q session and check that 20 events have been published to the console.

TASK 2 - Function Projections

TASK 4 - Parsing Messages (Part 1)