

# Q\_ENG DEV ver.2.0

JE KIM

2020-07-22

## 1. 개요

Q엔진 고도화 작업을 1차 완성하였습니다.

가장 주요하게 변경된 부분은 국내 주식 추가와 Input : 주간 종가 데이터, Output : 일간 종가 데이터 입니다.

그리고 누적 수익률 값을 fee와 transaction cost, Time Lag를 적용하여 자세하고 정확하게 계산하였습니다.

또한, 이전 버전의 Q엔진은 머니포트 서버에서 C로 임베딩하는 작업에서 호환이 어려운 문제가 있어 그 부분은 계속 수정 중에 있습니다.

## 2. 추가된 변수 설명

- ✓ **wght\_chg** : 비중의 변화량

리밸런싱할 때의 transaction cost를 계산하기 위해서는 비중의 변화량을 알아야 합니다.

- ✓ **wght\_chg\_sum** : [해외주식비중 변화량 합, 국내주식비중 변화량 합]

해외와 국내의 장 운영 시간이 다르기 때문에 매수하는 비중이 국내 주식에 더 많다면 Time Lag가 존재하기 때문에 필요한 변수입니다.

- ✓ **reb\_idx = -1** : 리밸런싱 순서를 나타내는 변수

첫번째 사이클에는 리밸런싱을 하지 않기 때문에 -1로 시작합니다. 따라서, 첫번째 리밸런싱일 때 변수의 값은 0이 됩니다.

- ✓ **tr\_b\_cost\_u** : 해외 거래 매수 비율(%)

- ✓ **tr\_s\_cost\_u** : 해외 거래 매도 비용(%)

- ✓ **tr\_b\_cost\_k** : 국내 거래 매수 비용(%)

- ✓ **tr\_s\_cost\_k** : 국내 거래 매도 비용(%)

- ✓ **l\_result** : sql에서 뽑은 한 종목 주별 증가 (lookback)

- ✓ **i\_result** : sql에서 뽑은 한 종목 일별 증가 (in\_dur)

- ✓ **l\_date** : lookback 주간 증가 순서와 같은 날짜 데이터 리스트

- ✓ **i\_date** : in\_dur 일간 증가 순서와 같은 날짜 데이터 리스트

- ✓ **rCost** : 종목별 주별 증가 = l\_result

- ✓ **tCost** : 종목별 일별 증가 = i\_result

- ✓ **Money\_sub** : 누적 수익률 서브 변수

구매 가능한 주식 수를 계산하는 Money 외의 거래세 비용을 차감하기 위해 필요한 매일 누적된 수익을 저장하는 변수입니다.

- ✓ **fee\_cum** : 누적 fee의 값

매일 fee를 (1/250)치를 지불하지만 Money에는 적용되지 않기 때문에 일별 수익률에 누적해서 차감해주어야 합니다.

- ✓ **tsc** : transaction cost 변수

- ✓ **excnt** : 환전 시차를 위한 날짜 카운트 변수

- ✓ **excnt\_x** : 환전 시차가 필요한 종목 유무 변수

### 3. 수정 완료 사항

#### 3.1 비중 - 주간 데이터 사용 / 수익률 - 일간 데이터 사용

기존에는 월간 데이터 사용으로 입력하는 개월 수만큼의 데이터만 불러오면 되었지만 주간과 일간 데이터를 사용하기 때문에 날짜의 중요성이 커졌습니다.

따라서, 날짜와 해당하는 데이터의 인덱스 값이 같은 `i_result`, `i_date` 변수를 생성했고 기간에 존재하는 날짜의 종가를 정확하게 찾아 비중과 수익률을 구하는데 사용했습니다.

#### 3.2 기대수익률 연/월 환산

기존에는 월간 데이터의 연 환산 수익률을 기대 수익률로 사용했습니다. Lookback 데이터를 주간 데이터로 변경 후, 연/월 환산 수익률로 비교를 해보았을 때, 결과로도출되는 비중 차이는 크지 않았습니다. 최종적으로는 연 환산 수익률을 사용하기로 결정되었고 월간에서 주간으로 바뀌었기 때문에 기존의 `data_dur_type` 값을 12에서 52로 변경하였습니다.

#### 3.3 국내 주식 도입

기존에는 해외 주식 데이터를 바탕으로 비중과 수익률을 그렸습니다. 하지만 곧 머니포트에 국내 주식까지 도입될 것을 대비해 국내 주식 데이터를 직접 데이터베이스에 import하였습니다. 국내 주식과 해외 주식의 구분은 국내 주식 티커가 'A + 숫자6자리' 임을 이용했습니다.

#### 3.4 환율 적용

국내 주식 데이터를 도입하여 함께 포트폴리오를 구성했기 때문에 해외 주식 데이터에 환율을 적용할 필요성이 있었습니다. 비중을 구하는 `rCost`는 환율을 적용하지 않았고 수익률을 구하는 `tCost`는 환율을 적용하여 누적 수익률을 구하였습니다.

### 3.5 Fee와 transaction cost 차감

Fee는 매일  $1/250$  (일년 동안의 영업 일수)만큼 차감합니다. 따라서, 누적되는 fee 값을 저장하는 변수가 필요했습니다. 또한, 매 날짜의 fee는 그 전날까지의 누적된 fee가 차감된 money값이어야 하므로 그 부분에 유의하여 프로그래밍하였습니다. 초반에는 누적 fee라는 점에 집중하여 각 일간 누적 수익률 데이터에  $-1/250$ ,  $-2/250$ , ... 차감하는 공식을 생각했지만 이는 fee를 최대로 누적하는 방법이기 때문에 옳지 않았습니다. 또, 누적 수익률이기 때문에  $-1/250$ ,  $-1/250$ , ... 공식도 옳지 않습니다. 따라서,  $-1/250$ ,  $-1/250-a$ ,  $-1/250-b$ , ... 공식을 이용해 누적 Fee를 구했습니다.

Transaction cost는 매수/매도가 이루어질 때 발생하는 비용입니다. 첫 매수와 마지막 매도, 그리고 비중이 변화될 때 발생하게 됩니다. 비중이 변화할 때를 고려하는 것이 까다로웠는데 변화한 비중의 +/-에 따라 매수/매도를 구분하여 차감해주었습니다.

각 날짜를 엑셀과 비교하며 민감하게 인덱스 처리를 해주었습니다. 그래서 코드의 순서가 매우 중요하므로 후에 수정이 필요할 때 참고를 위해 작성한 엑셀이 유용할 것입니다.

### 3.6 국내/해외 주식 운영 시간 차이 Time Lag 적용

국내와 해외의 시차는 존재합니다. 국내 증시는 오전 9시부터 낮까지 운영이 되고 해외 증시는 우리나라 시간으로 오후 저녁부터 새벽까지 운영이 됩니다. 따라서, 국내 주식 비중 변화량 합과 해외 주식 비중 변화량 합을 비교했을 때, 해외 주식 비중이 늘어났다면 국내 주식을 먼저 매도한 후, 그의 증거금을 통해 해외 주식을 매수하는 것이 가능합니다. 하지만 국내 주식 비중이 늘어났다면 해외 주식을 먼저 매도한 후에 다음 날에 국내 주식을 매수해야 하므로 당일에 거래가 모두 이루어지는 것은 불가능합니다. 따라서, 1일의 Time Lag가 발생하여 이를 적용해주었습니다.

환전 시차가 존재하는가에 대해서도 고려를 했습니다. 거래가 완료되는  $T+2$ 일과 환전이 이루어지는  $+1$ 일, 총  $T+3$ 일 환전 시차가 있을 것이라고 예상하여 3일의 Time Lag를 두었습니다. 하지만 실제로 대표님과 직접 거래를 진행해보고 한국투자증권에 문의해본 결과, 최근에는 대부분의 증권사들이 증거금으로 당일 매수가 가능하게 시스템을 갖췄다는 것을 확인했습니다. 따라서, 위의 내용을 바탕으로 3일 Time Lag를 둔 것을 1로 변경하였습니다.

### 3.7 소수점 매매 불가

신한금융투자는 소수점 매매가 가능하지만 KB증권은 소수점 매매가 불가합니다. 기존에 소수점 매매가 가능했던 코드를 변경하여 일의 자리 수 매매만 가능하게 변경하였습니다.

### 3.8 성과 분석 추가

자문사의 전략팀으로부터 전달받은 PortfolioAnalysis\_HT.py 파일을 import 하여 마지막 도출된 일자 별 누적 수익률을 대입해 성과 분석을 했습니다. Q엔진의 Day는 list 형태로 print되기 때문에 이를 dataframe으로 변환하여 대입했습니다. 결과물은 html파일로 생성되어 보여집니다.

## 4. 예외 처리

### 4.1 국내와 해외 휴업일 차이로 인한 오류

국내와 해외의 휴일이 다르기 때문에 누락되는 날짜가 있어서 날짜 인덱스가 일치하지 않는 오류가 있었습니다. Q엔진은 우선 국내 이용자를 대상으로 하는 엔진이기 때문에 국내 휴업일에 해당하는 날짜는 삭제했습니다. 반면, 해외 휴업일에 해당하는 날짜는 그 전날 종가 데이터로 채웠습니다.

### 4.2 C 임베딩을 위한 self와 return 수정

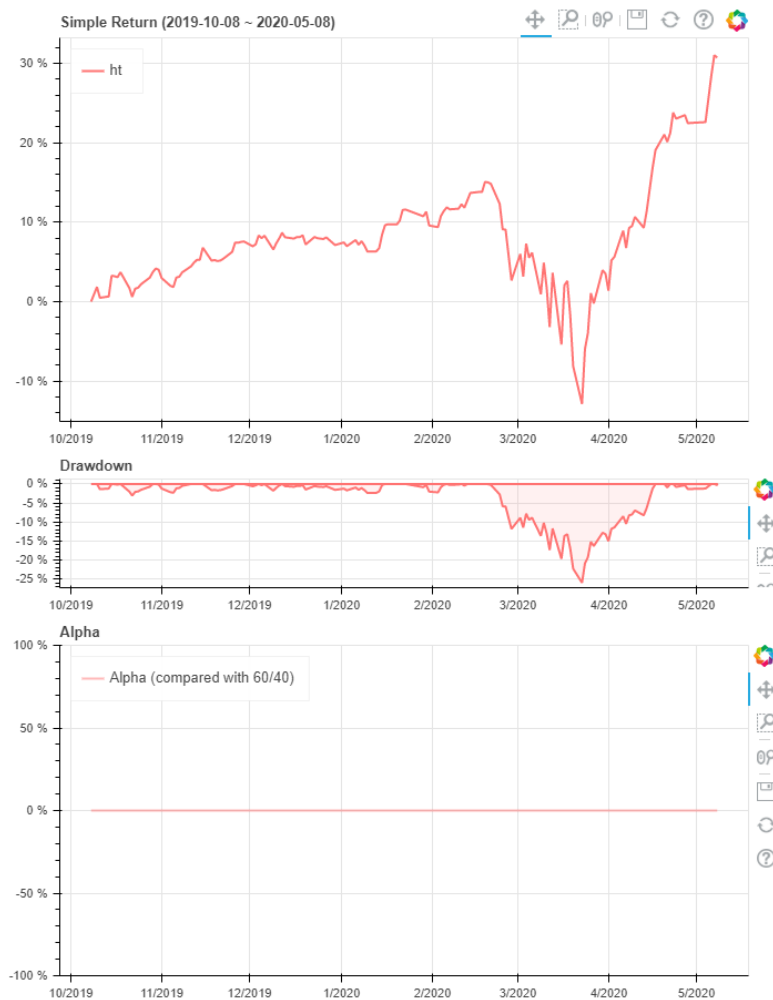
Moneypot의 서버에 올리기 위해서는 C 임베딩 작업이 필요합니다. Q엔진은 파이썬으로 개발되어 있기 때문입니다. 하지만 C에서는 파이썬의 self를 인식하지 못하는 이슈가 있었습니다. 이에 대해 플랫폼 개발팀과 이야기한 결과, self를 제거하고 return을 정확하게 받게 다듬은 전체 코드를 플랫폼팀에 전달할 필요성이 있습니다.

### 4.3 (미완료) 신규 상장 종목

신규 상장 주식은 과거 종가 데이터가 충분하지 않기 때문에 SQL 에러가 생길 것입니다. 이를 사용자에게 그대로 보여줄 수 없으므로 예외처리 방법에 대해 고민이 필요합니다.

신규 상장 종목을 선택할 시에는 사용자에게 알림을 보내는 방향으로 진행할 예정입니다.

## 5. 실험 결과



| # |                    | ht      |
|---|--------------------|---------|
| 0 | Simple_Return      | 30.73%  |
| 1 | CAGR               | 60.17%  |
| 2 | Sharpe Ratio       | 1.9933  |
| 3 | Sortino Ratio      | 3.2862  |
| 4 | Standard Deviation | 35.31%  |
| 5 | MDD                | -25.99% |
| 6 | Average Drawdown   | -3.66%  |

## 6. 결론

Q엔진이 전체적으로 많은 부분이 변경되었습니다. 우선적으로 국내 주식 도입과 입력과 출력 표현 데이터가 달라졌기 때문에 그 사이에 계산 과정이 크게 수정되었습니다. 국내 주식과 해외 주식을 구분하기 위한 연산과 날짜를 구분하기 위한 start, end 인덱스 연산이 기본으로 비중과 수익률을 구하는 공식에 포함되었습니다.

가장 고민을 많이 했던 부분은 Fee, Transaction cost 부분입니다. 기존 코드에는 매 월 리밸런싱을 진행하는 방향으로 작성되어 있었기 때문에 Fee를 어떻게 누적하는 것이 가장 정확한지 생각을 많이 했습니다. Transaction cost는 코드 구현 방법이 가장 난제였습니다. 어느 날짜에 비용이 발생하는지가 가장 중요했기 때문에 인덱스에 맞는 정확한 계산을 위해 print를 여러 번 출력해보며 엑셀과 비교 검증했습니다. 첫번째 비중일 때, 중간 비중일 때, 마지막 비중일 때의 코드가 모두 상이합니다. 그래서 Q엔진의 코드 순서가 매우 중요하며 향후 수정할 시, 인덱스를 고려한 수정이 필요합니다.

Time Lag 구현으로 인해 코드가 복잡해지고 길어졌습니다. 해외 주식 비중 변화량의 합과 국내 주식 비중 변화량의 합을 비교해야 하고 그 안에서도 매수/매도 시차에 관한 고려가 필요했기 때문입니다.

완성된 Q엔진 파일을 플랫폼 개발팀에 전달할 때에는 호환이 편한 파일을 전달해야 합니다. Q엔진은 서버에서 작동되기 때문에 필요한 예외처리는 파이썬에서 처리해서 전달하는 것이 맞다고 생각합니다. 따라서, 전달될 최종 파일은 좀 더 정리되어서 보내질 예정입니다.