Rebalancing, Weight, Mon

JE KIM

2020-06-24

1. 개요

Q_eng의 최종 비중을 도출 후에, 이에 의한 누적 수익률을 구하는 과정을 분석하였습니다.

결괏값은 각 월마다의 누적된 수익률 리스트인 Mon입니다.

Q_eng에서 출력되는 값은 누적 수익률 또는 누적 자산값이며 모두 fee를 제외한 값입니다. 검증되는 기간은 과거의 기간으로 최종 비중은 리밸런싱 마지막 비중 값으로 도출됩니다.

2. 변수 설명

lookback	비중 구하는 기준 기간
dur	수익률 검증 기간 (백테스트)
Money	총 자산
Mon	그래프 그릴 누적 자산 값 (룩백 기간 + 입력 기간)
rebal	리밸런싱 값 리스트 ex) [3,6,9]
weight	비중 값
wght_hst	월마다의 비중 값

입력한 dur 기간동안 비중에 따른 수익률을 검증합니다. 리밸런싱할 개월 수는 한 개 기준으로 프로그래밍 되어 있지만, 리스트 형식으로 여러 개 지정 가능합니다. 입력한 개월마다 리밸런싱을 통해 비중을 새롭게 도출하고 그로 인한 누적 수익률을 그려냅니다.

Money는 단일 변수이고 Mon은 입력 기간(in_dur) 동안의 리스트 변수입니다. 따라서, Mon의 0번째 인덱스에는 초기 수익률인 1이 위치하게 됩니다. 투자 금액을 1로 설정 시, 누적 수익률을 확인할 수 있고, 원하는 금액으로 설정 시에는 누적 자산 값을 확인할 수 있습니다.

코드 중간에 존재하는 lap은 매 월의 누적 수익률을 구하기 위한 월 단위를 뜻하는 변수입니다.

Weight은 최종 비중 값이며 wght_hst는 리밸런싱을 통한 비중의 변화를 확인할 수 있는 매 월마다의 비중 값입니다. 둘 다 마지막에는 리밸런싱 리스트의 마지막 값 기준으로 출력됩니다.

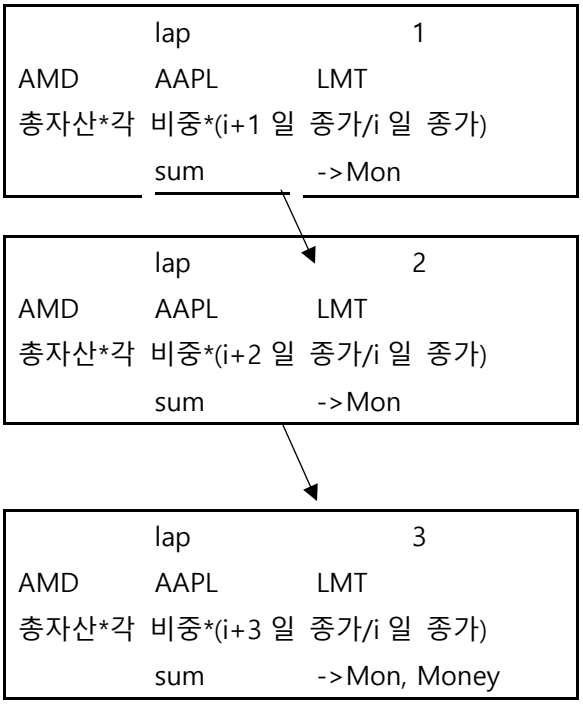
3. 누적 수익률 도출 과정

입력 값

lookback 기간(개월 수): 20
투자 금액: 1
비중배분 알고리즘 선택(1: Markowiz, 2: BlackLliterman): 1
rebalancing dur(ex: [3,6,9]) : [3,6]
기간 입력(개월 수): 7
종목 코드 입력(ex: [ABC, DEF, HIJ])): AMD,AAPL,LMT

reb 3

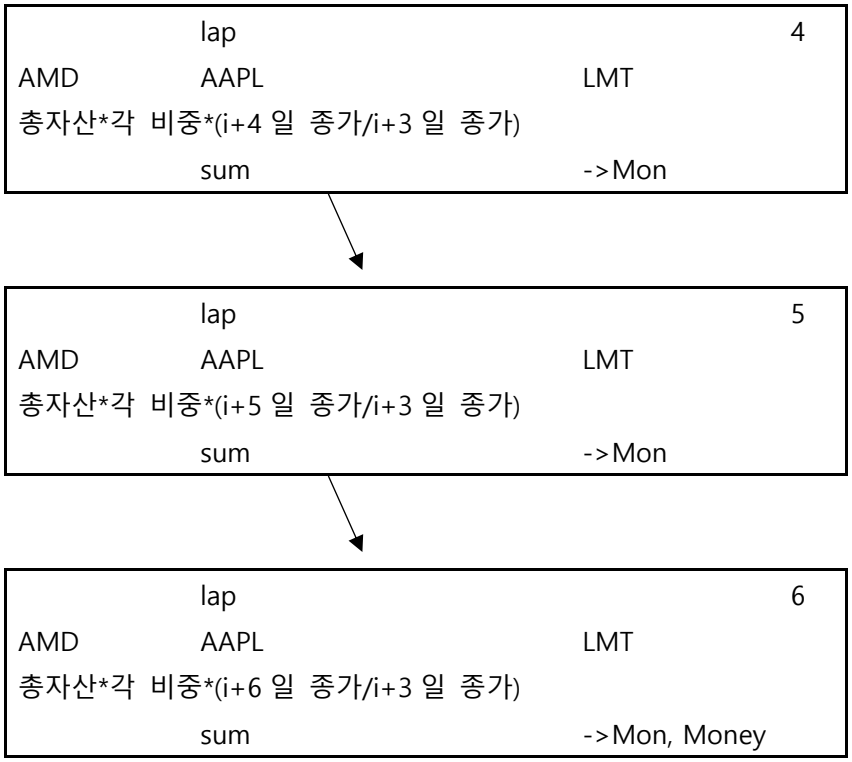
lookback 동안 수익률을 통해 비중 찾기
그 비중을 통한 누적 수익률, 6 개월동안 검증 시작
3 개월마다 리밸런싱



Mon		20		3	3

첫번째 금액은 초기 금액인게 맞음

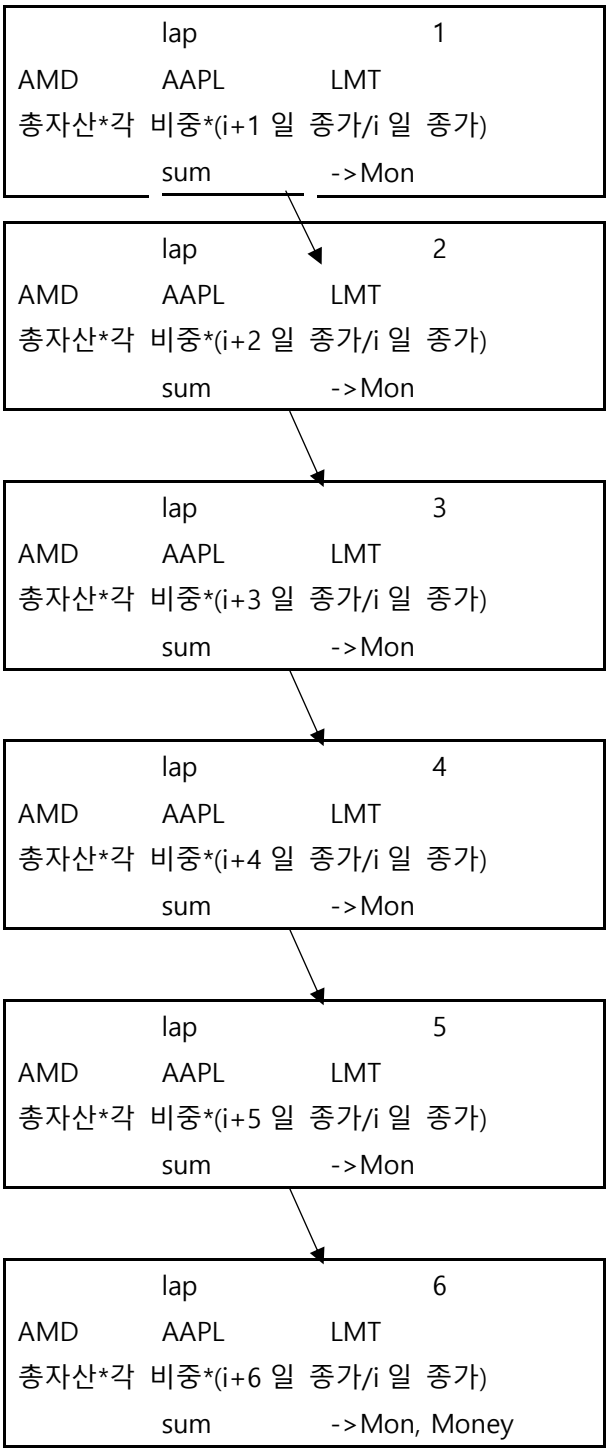
다시 그 직전 lookback 동안 수익률을 통해 비중 찾기(리밸런싱)



Mon		23			3

reb **6**

lookback 동안 수익률을 통해 비중 찾기
그 비중을 통한 누적 수익률, 6 개월동안 검증 시작
6 개월마다 리밸런싱이므로 리밸런싱 없음



Mon		20		6

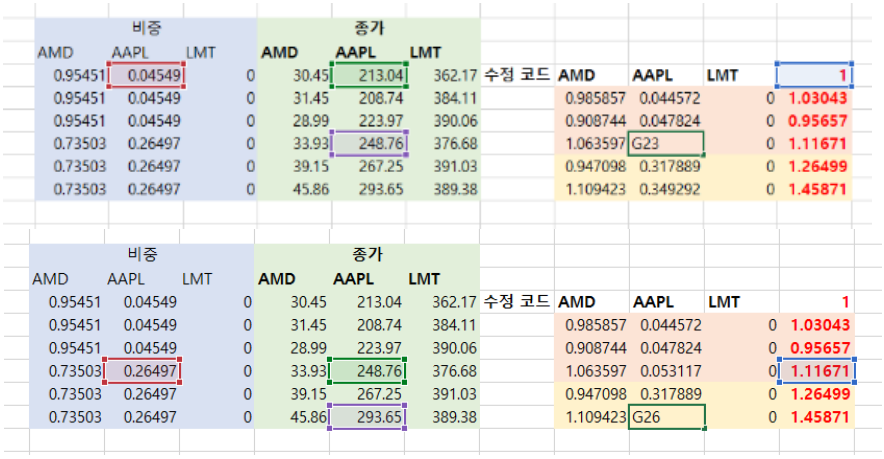
첫번째 금액은
초기 금액인게 맞음

4. 수정 사항

- Mon의 리스트 형태를 만드는 과정에서 (리밸런싱 값의 수 + 1) 만큼 리스트가 만들어지는 문제를 리밸런싱 값의 수만큼 만들어지도록 수정했습니다.

- 입력한 검증 기간이 입력된 리밸런싱 값의 배수가 아니면 인덱스 에러가 발생하는 문제를 break문을 추가하여 수정했습니다.

- 복리 방식으로 누적 수익률을 구하는 과정으로 구현되어 있어 *종가 수익 계산 방법*과 *총 수익률 변수 값*을 수정했습니다. 아래는 입력 기간 6으로 설정 시의 결과입니다.



비중			종가							
AMD	AAPL	LMT	AMD	AAPL	LMT	수정 코드	AMD	AAPL	LMT	1
0.95451	0.04549	0	30.45	213.04	362.17		0.985857	0.044572	0	1.03043
0.95451	0.04549	0	31.45	208.74	384.11		0.908744	0.047824	0	0.95657
0.95451	0.04549	0	28.99	223.97	390.06		1.063597	G23	0	1.11671
0.73503	0.26497	0	33.93	248.76	376.68		0.947098	0.317889	0	1.26499
0.73503	0.26497	0	39.15	267.25	391.03		1.109423	0.349292	0	1.45871
0.73503	0.26497	0	45.86	293.65	389.38					

비중			종가							
AMD	AAPL	LMT	AMD	AAPL	LMT	수정 코드	AMD	AAPL	LMT	1
0.95451	0.04549	0	30.45	213.04	362.17		0.985857	0.044572	0	1.03043
0.95451	0.04549	0	31.45	208.74	384.11		0.908744	0.047824	0	0.95657
0.95451	0.04549	0	28.99	223.97	390.06		1.063597	0.053117	0	1.11671
0.73503	0.26497	0	33.93	248.76	376.68		0.947098	0.317889	0	1.26499
0.73503	0.26497	0	39.15	267.25	391.03		1.109423	G26	0	1.45871
0.73503	0.26497	0	45.86	293.65	389.38					

[[1, 1.03, 0.957, 1.117, 1.265, 1.459]] → 코드의 수수료를 제외한 값과 일치 확인

5. 결과

Mon:
[[1, 1.003, 1.036, 0.955, 1.089, 1.215, 1.383],
 [1, 1.003, 1.036, 0.955, 1.117, 1.289, 1.51]]
Weight :
[[63.933]
 [36.067]
 [0.   ]]
Wght_hst :
[[99.997, 0.003, 0.0], [99.997, 0.003, 0.0], [99.997, 0.003, 0.0], [99.997, 0.003, 0.0],
 [99.997, 0.003, 0.0], [99.997, 0.003, 0.0], [63.933, 36.067, 0.0]]

fee 제외

6. 결론

전체적인 흐름은 룩백 기간과 입력 기간의 데이터를 이용하여 비중을 통한 누적 수익률을 확인하고 최근 룩백 기간을 통해 새롭게 비중을 도출해내는 방식입니다. 비중을 확인하고 싶다면 리밸런싱 값은 한 가지 값만 입력하는 것을 지향합니다. 리밸런싱 값에 따른 누적 수익률을 비교하고 싶다면 리밸런싱 값에 여러 값을 입력하여도 좋습니다.

파이참에서 실행하였을 때의 코드 속도와 웹에서 불러왔을 때의 속도를 확인하였을 때, 아이라에 적용한다면 전보다 속도 측면에서의 성능도 개선될 것이라고 생각합니다.