

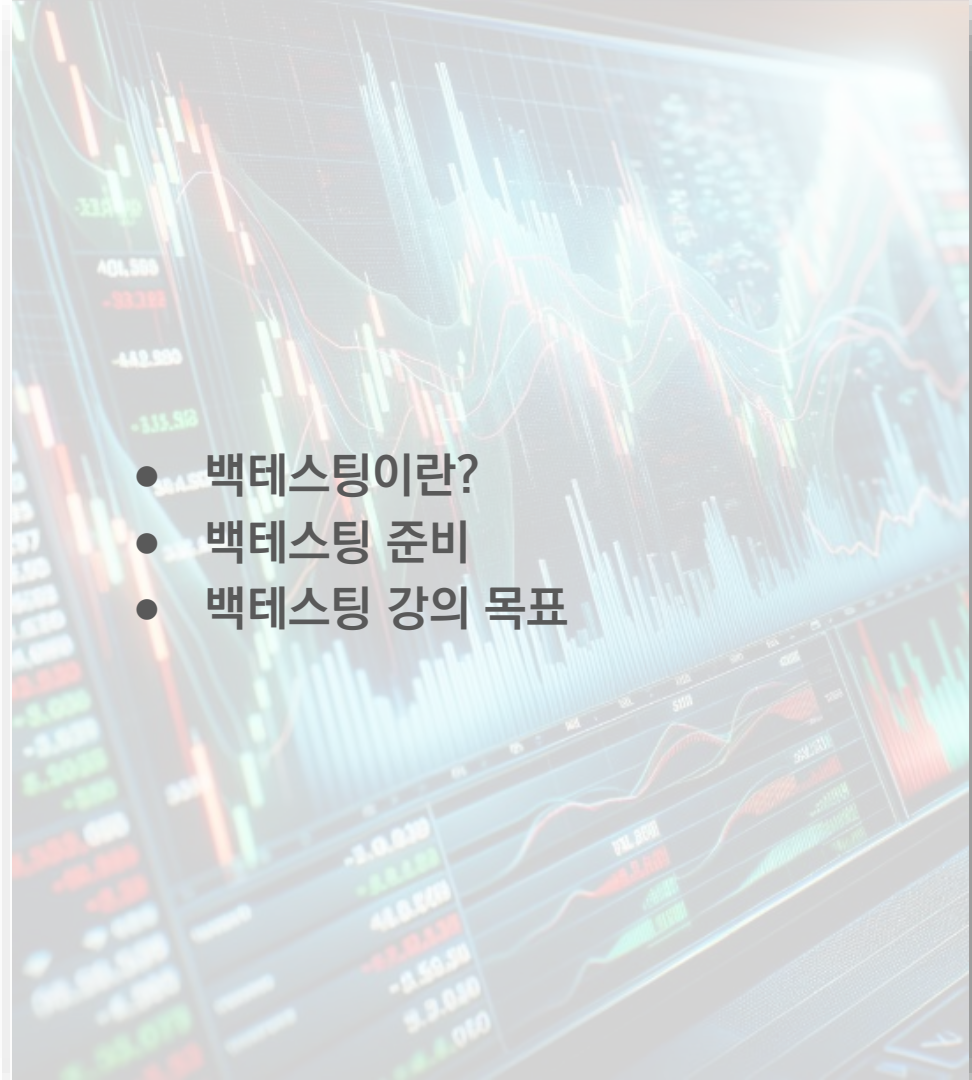
알고리즘 트레이딩 정복하기



4장. 전략 백테스팅

전략 백테스팅 개요

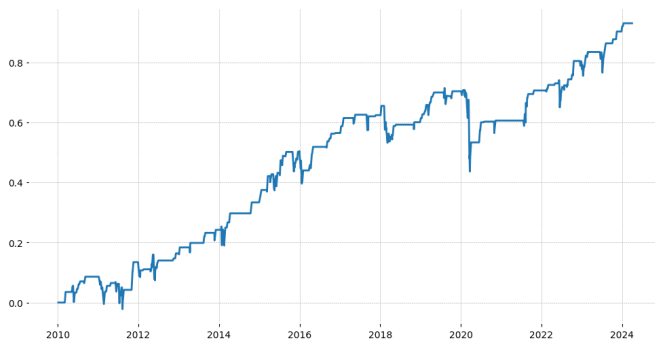
- 백테스팅이란?
- 백테스팅 준비
- 백테스팅 강의 목표



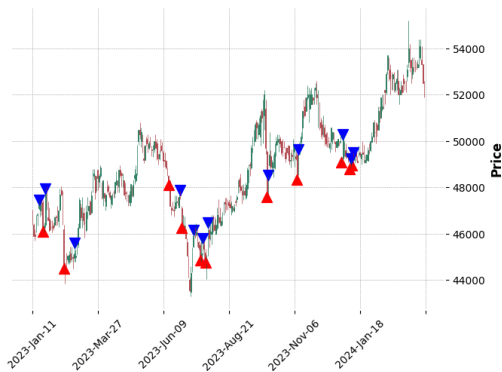
백테스팅이란?

● 백테스팅(Backtesting)

- 과거 데이터(back)로부터 내 전략의 예상 수익과 리스크를 정량적으로 평가해볼 수 있는 테스트(Testing) 방법



전략의 과거 기간 수익률 확인



내 전략의 매수/매도 타점 확인

백테스팅 정의

● 왜 백테스팅을 할까?

- 전략을 개발할 땐 두 가지 의문이 생김
 - 정말 수익이 날까?
 - 여러 전략(RSI 10 미만 종목 매수해 10/20/30일 후 팔기) 중 어느게 제일 좋을까?
- 이 의문을 해결하기 위해 백테스팅이 존재

● 모든 전략은 백테스팅에서 시작

- 대부분의 퀀트 전략은 무조건 백테스팅을 진행
- 가장 유명한 퀀트 헤지펀드 르네상스 테크놀로지스는, 백테스팅 기법을 극한으로 다듬어 좋은 전략을 잘 구분하게 된 것이 첫 번째 성공 비결
- 퀀트 전략 개발에 있어 가장 중요한 부분

백테스팅 준비 - 준비물

- 백테스팅 준비물은 전략 아이디어, 데이터

전략

RSI가 40 이하인 종목을 매수해서 5일 후에 팔자

데이터

RSI 데이터: 매수 타이밍을 계산하기 위해 필요

주가 데이터: 매수/매도 가격을 계산하기 위해 필요

- 좋은 품질의 데이터 수집이 가장 중요

- 알고리즘 매매는 수천/수만번 매매하기 때문에, 거래내역을 하나하나 분석하기 어려움
- 따라서, 데이터에 빈 날짜가 있거나 잘못된 값이 입력되는 등, 낮은 품질의 데이터를 사용하면 잘못된 백테스팅 결과로 유도

백테스팅 준비 - 데이터셋 분할

- 백테스팅 데이터는 Train/Test 두 구간으로 나눔
 - Train 데이터로 전략을 개발, 개발한 전략을 Test 데이터로 최종 테스트



- Train: 여러가지 전략 조합 중, Train 기간에서 성과가 잘 나오는 조합 탐색
- Test: 선택한 최적 전략의 Test기간 성과와 Train기간 성과를 비교
 - Train에서 선택한 전략이 Test 기간에서도 잘 작동한다면, 전략의 신뢰도 상승
 - Train기간과 다른 흐름을 보인다면, 어떤 요인으로 인해 다른 흐름을 보이는지 분석

Slippage란?

- 백테스팅의 문제점 – 백테스팅은 완벽하지 않다!
 - 백테스팅에서 매매 시점의 호가를 정확히 알기 어려움
 - 거래대금이 작은 종목을 거래할 땐 내 주문으로 호가/체결가격이 변할 수 있음
 - 이 때문에, 백테스팅과 실거래의 수익률이 다를 수 있음
- Slippage 개념으로 문제 보완
 - 백테스팅 수익률을 보수적으로 측정하기 위해, 매매대금의 일정 부분을 비용처리(Slippage)
 - 이같은 간단한 작업으로 모델 신뢰도를 크게 높일 수 있음

$$\text{Slippage} = 0.2\%(\text{수수료} + \text{세금}) + 0.2\%(\text{추가 비용 가정}) = 0.4\%$$

용어 정리

- 미래참조 – 오늘 매매 결정 과정에 오늘 이후 시점(미래)의 데이터가 개입
 - 미래 일어날 일을 알고 투자하는 셈이므로, 백테스팅 수익률은 높음
 - 실거래에선 미래 데이터를 미리 알 수 없기에 오히려 손실 발생
- 생존편향 – 현재 시점의 상장 종목만으로 백테스팅
 - 이미 좋지 않은 사업 결과로 사라진(상폐된) 종목은 제외하기 때문에, 백테스팅 수익률은 높음
 - 실거래에선 어떤 종목이 미래에 살아남을지 알 수 없어, 백테스팅보다 수익률 크게 낮아짐

용어 정리

- 오버피팅- 좋은 결과가 나올 때까지 백테스팅을 반복 실행
 - 백테스팅을 무한 반복하면 때때로 과거 잘 통했던 전략 발견
 - 하지만, 이 전략이 미래에도 잘 통한다는 보장 없음. 대부분은 통하지 않음
 - 따라서, 오버피팅 전략은 대부분 실거래 직후 손실 발생

4.1

백테스팅 강의 목표

백테스팅 강의 목표

- 내 전략 아이디어가 유의미한 전략인지 확인
- 백테스팅 강의 수강 후 할 수 있는 것
 - 내 전략을 프로그램으로 구현
 - 내 전략의 과거 수익률 그래프 그리기
 - 내 전략의 매수/매도 타점 그리기

```
pos = 0
cum_rt = []

bidpoint = []
askpoint = []

for i,r in d.iterrows():
    cum_rt.append((r['close']/r['lastc
    askpoint.append(np.nan)
    bidpoint.append(np.nan)
    if r['rsi'] < 40 and pos == 0:
        pos = 1
        bidpoint[-1] = 1
    elif r['rsi'] > 40 and pos > 0:
        pos = 0
        askpoint[-1] = 1

plt.figure(figsize=(15,6))
plt.plot(d.index,pd.DataFrame(cum_rt).
```

