파이썬으로 똑똑하게 주식하기

Python이 가져다 줄 **황금알을 낳는 거위는 있을까요?**



김대현(Damon)

reinitiate@gmail.com admin@financipe.com www.financipe.com

발표자 소개



김대현

reinitiate@gmail.com admin@financipe.com www.financipe.com/me @fb/reinitiate

量洲 伯色, 品品計, ETF, 品知的时, 化树色的时, Pyconzols Lightening talk, 파이松, 스儿, 외州科외 珠柳이더 荆스트, 希伯里 至的故.

발표자의 파이썬 사랑





PYTHON#31577

주식시장의 개인투자자 (1)





[서울경제TV] 상승장에도 개인투자자 매수종목 역주행 서울경제 | 2시간 전 | 네이버뉴스 | []

상승장에서도 여전히 **개인투자자**들만 울상을 짓고 있는 것으로 나타났습니다. 이보경기잡니다. [기자] 금융 정보업체 에프앤가이드에 따르면 개인 순매수 상위 10종목의 7월 1일 대비 전날 주가는 평균 11.72% 하락 한...



[데이터뉴스]코스피 **개인 투자자** 가장 많이 사들인 10종목 중 9곳 손실

이투데이 | 9시간 전 | 🗹

이투데이=유혜은 기자 | 올 하반기 상승세를 보이는 유가증권시장(코스피)에서 **개인투자자**들만 손실을 본 것으로 나타났다. 11일 금융정보업체 에프앤가이드에 따르면 코스피 투자 주체별 순매수 상위 10종목의...

└ 코스피 상승에도 **개인투자자** 순매수 종목은··· 한국경제TV │ 10시간 전 │ 네이버뉴스

<u>뛰는 주가, 구경만 하는 개인투자자</u> 중앙일보 | 🗰 E5면1단 | 1일 전 | 네이버뉴스 | 🚅

하지만 **개인투자자**들은 주식 시장에 추가로 들어갈 기미가 없다. 이날 금융투자협회에 따르면 증시 대기자금인 투자자 예탁금은 8일 현재 22조5803억원으로 지난달 초(25조4615억원)보다 3조원 가량 줄었다....

WHY?

주식시장의 개인투자자 (1)





개인투자자는 왜 항상 실패하는가?

- o 기관/외인에 비하여 정보량 부족
- o 감에 의존한 묻지마 투자
- o 불확실한 정보에 투자
- o 자신만의 물이 없는 투자
- o 비이성적인 행동 : 원금회복에 대한 막연한 기대
- o 남들이 간다고 따라가는 **맹목적 투자**
- 지속성 없음
 값ば에 축가ば니다.

주식시장의 개인투자자 (2)



그럼 주식투자를 하지 말아야 하나요?

- o 연 1-2% 초저금리 시대: 천만원 예금하면 10만원 받습니다.
- o 물가상승률 〉 금리: 예금은 곧 손실과 같습니다.
- o 30년 저축하면 강북에 집 살수 있는 대한민국
- o 보통 사람이 자본이득을 취할 수 있는 기회
- o 데이터 확보/활용의 진입장벽이 매우 낮아짐
- o 파이썬이라는 생산성 높은 언어로 인하여 개인도 기관/외인에 대항할 수 있는 정보획득 가능

그렇다면 주식투자 하되 똑똑하게 하자! 파이썬으로!

시스템 트레이딩 (1)



- 정보/소문이 아닌 특정한 전략(룰)을 그대로 실천하는 전략
- 계량적인 전략기반의 투자를 하면 비이성적인 판단에서 벗어날 수 있음
- 성장하는 ETF/ETN 패시브 투자 상품도 일종의 시스템 트레이딩 (제도권 트렌드)
- 시스템 트레이딩의 3단계
 - 데이터 확보
 - 전략개발 및 검증
 - 이행 (주문 / 계좌관리)
- 꼭 자동으로 주문을 넣지 않아도 룰에 기반한 트레이딩을 한다면 OK
- 지속되는 특정한 현상 혹은 자신의 아이디어를 수익으로 바꿔내는 일
- 파이썬은 시스템 트레이딩에 가장 좋은 언어

시스템 트레이딩 (2)



Python을 통한 시스템 트레이딩으로 개인투자자의 문제를 해결 가능한가?

비이성적인 대처	정보량 부족	불확실한 정보	지속성 없음
룰에 의하여 자동으로	여러 스크래핑 라이브	정확한 데이터를 계량	룰베이스로 투자하기
매수/매도를 하므로 비	러리 혹은 증권사 API를	적으로 가공하여 매수/	때문에 정성적인 투자
이성적인 행위 불가	통해서 수취 가능	매도 신호 생성	방식보다는 지속됨
완전 해결가능	다소 해결가능	해결가능	시장은 변함

왜 파이썬인가?



1) 시스템 트레이딩의 3요소가 한개의 언어로 커버

데이터 확보

- Web data extraction
 - Beautiful-Soup
 - Reqeust, StringIO
 - 가장 짧은 스크래핑 소스
- MysqlDB
 - 간단한 관계형 데이터베이스 사용

로직 (분석)

- Numpy, pandas
 - 관계형 데이터베이스 형식의 숫자 를 다루는 강력한 라이브러리
- Statsmodel
 - 통계, 머신러닝
- Matplotlib
 - 강력한 시각화 도구

이행(주문 및 계좌관리)

- 증권사 API
 - win32com COM object 활용
- 다양한 엑셀관리 라이브러리
 - 엑셀형태의 레포트
- Flask
 - 간단하고 빠른 웹 개발

2) 생산성

- 시스템 트레이딩 특성상 다수의 인원이 협업하기 어려움. (로직 유출)
- 1인이 기획/서버/ 클라이언트 전부 개발해야 하는 상황이 빈번
- 결국 투입시간대비 얼마나 생산성을 끌어내느냐가 관건
- 이런 관점에서 파이썬은 매우 뛰어난 언어



전체적인 프로세스를 한바퀴 돌아봅시다

IDEA



파이썬을 다룰줄 아는 주식시장에 관심이 많은 투자자 A씨는 다음과 같은 생각이 들었다

- " 오른 주식이 더 오르는 것 같은데?"
- "9개월동안 가장 많이 오른 주식 15개를 똑같은 비중으로 들고가면 어떨까?"
- " 주식은 매달 말에 교체하는 것이 어떨까?

A씨는 자신의 아이디가 옳은지 검증해본 후에 실제 투자 여부를 결정하기로 하였습니다.



데이터 탐색



목적을 이루기 위해 어떤 데이터가 필요한가?

카테고리	항목	수급처	필요여부
I J-77 L	국내 영업일	종목일별데이터	O
날짜	해외 영업일	종목일별데이터	X
	종목코드(A00000)	<u>www.financipe.com/api/</u> stockmaster	O
종목 마스터	종목이름	<u>www.financipe.com/api/</u> stockmaster	O
	업종	http://finance.naver.com/	X
	종가	http://finance.naver.com/	O
	시가		X
종목	고가		X
일별데이터	저가		X
	수익률	http://finance.naver.com/	O
	거래량 (거래대금)	http://finance.naver.com/	O

데이터 확보 - 종목코드(1)



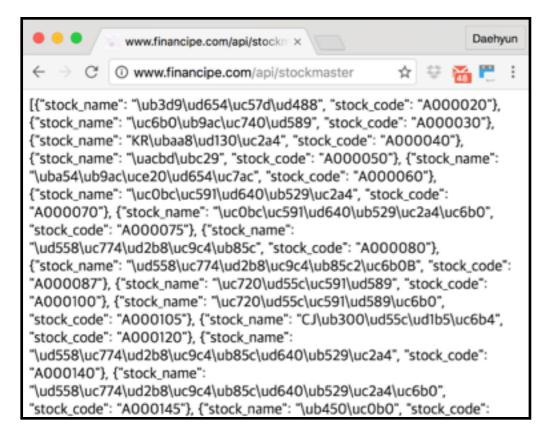
- 종목코드
- 약 2000개의 상장기업 코드
- A000000 형태로 존재
 - 보통주: 끝이 0으로 끝남
 - 우선주: 끝이 0이 아님
- 기업 이름도 함께 적재
- 리츠, 부동산투자회사, 선박투자회사 같은 경 우에는 제외
- 데이터소스
- 대신 사이보스 트레이더 API
- www.financipe.com/data/master
- · 완료여부는 Slack을 통해서 로그 확인

```
import win32com.client
     import conn # DB connection
     import slacker
     def insert(conn, stock_code, stock_name):
         cur = conn.cursor()
         saltext = """
             insert into stock_mast(stock_code, stock_name)
             values('%s', '%s')
9
         """ % (stock code, stock name)
10
11
         cur.execute(sqltext)
12
         conn.commit()
13
         cur.close()
14
15
     def send_log_to_slack(user_name, msg):
16
         ts = datetime.datetime.now().strftime('%H:%M:%S')
17
         msg = '%s `%s` %s'%(ts, msg)
18
         slack = Slacker(token)
         slack.chat.post_message(channel, msg, username=user_name)
19
20
     cp_code_mrg = win32com.client.Dispatch("CpUtil.CpCodeMgr")
21
     stock_code_list = cp_code_mrg.GetStockListByMarket(1)
22
23
24
     name_dict = {}
25
     for stock_code in stock_code_list:
27
         stock name = cp code mrg.CodeToName(stock code)
         insert(conn, stock_code, stock_name)
28
29
     send_log_to_slack('적재', '종목마스터입력완료 총 %s건'%(len(stock_code)))
```

데이터 확보 - 종목코드(2)



- Financipe 종목코드 API 활용
 - www.financipe.com/api/stockmast
 - · JSON 형태로 반환



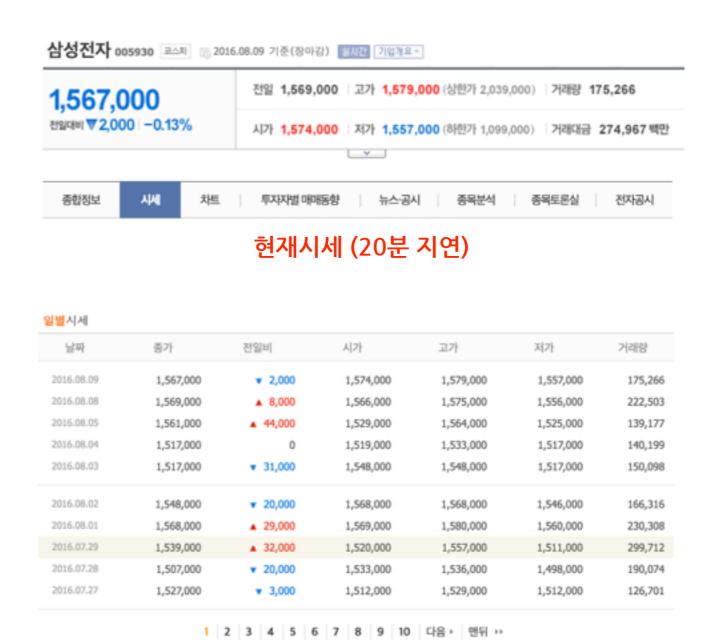
- urllib.urlopen(url)
- parse json string = json.load()

```
# -*- coding: utf-8 -*-
     import conn # DB connection
     import slacker
     import datetime
     import json
     import urllib
     from StringIO import StringIO
     from bs4 import BeautifulSoup
10
     def insert(conn, stock_code, stock_name):
11
12
         cur = conn.cursor()
         saltext = """
13
14
             insert into stock_mast(stock_code, stock_name)
             values('%s', '%s')
15
         """ % (stock_code, stock_name)
16
17
         cur.execute(sqltext)
18
         conn.commit()
19
         cur.close()
20
21
22
     def send_log_to_slack(user_name, msg):
23
         ts = datetime.datetime.now().strftime('%H:%M:%S')
24
         msg = '%s `%s` %s'%(ts, msg)
25
         slack = Slacker(token)
         slack.chat.post_message(channel, msg, username=user_name)
26
27
28
29
     data = urllib.urlopen('http://www.financipe.com/api/stockmaster')
     io = StringIO(BeautifulSoup(data, "lxml").text)
30
31
     items = json.load(io)
32
33
     for item in items:
34
         stock_code = item['stock_code']
         stock name = item['stock name']
35
36
37
         insert(conn, stock_code, stock_name)
38
     send_log_to_slack('적재', '종목마스터입력완료 총 %s건'%(len(items)))
39
```

데이터 확보 - 종가 및 수익률



- 데이터 소스
 - http://finance.naver.com/item/ sise.nhn?code=stock_code
- 필요한 데이터 (이력데이터)
 - 좋가
 - 거래대금
 - 거래량
- 준실시간 데이터
 - 현재가
 - 거래량
- · Request, Beautiful-Soup



일별시세

동시에 영업일 데이터까지!

데이터 확보 - table schema



```
CREATE TABLE 영업일테이블
   날짜 DATE PRIMARY KEY,
   전일날짜 DATE,
   휴장일구분 CHAR(1),
   외국휴장일구분 CHAR(1),
   요일구분 CHAR(1) NOT NULL,
   월별마지막영업일구분 CHAR(1),
   주별마지막영업일구분 CHAR(1)
);
CREATE TABLE 종목마스터
   종목코드 CHAR(7) PRIMARY KEY,
   종목이름 VARCHAR(20)
);
CREATE TABLE 일별시세
   날짜 DATE PRIMARY,
   종목코드 CHAR(7) PRIMARY KEY,
   시가 INT,
   고가 INT,
   저가 INT,
   종가 INT,
   수익률 DOUBLE,
   거래량 INT,
   거래대금 INT
);
```

확보완료 테이블

```
아이지어를 챖나라 정보 만들기!
CREATE TABLE 포트폴리오
   포트폴리오코드 VARCHAR(20) PRIMARY KEY
   리밸런싱일자 DATE PRIMARY,
   종목코드 CHAR(7) PRIMARY KEY,
   비중 DOUBLE
   진입구분 VARCHAR(1) -- 0:전일종가, 1:당일시가
);
CREATE TABLE 프토폴리오_성과
   포트폴리오코드 VARCHAR(20) PRIMARY KEY,
   날짜 DATE PRIMARY, KEY,
   수익률 DOUBLE
);
CREATE TABLE 프토폴리오_지표
   포트폴리오코드 VARCHAR(20) PRIMARY KEY,
   초기날짜 DATE,
   마지막날짜 DATE,
   연평균수익률 DOUBLE,
   변동성 DOUBLE
   승률 DOUBLE,
   MDD DOUBLE -- Maximum drawdown
);
```

테이블을 세개 더 만듭니다.

로직 - 과거 9개월 수익률 적재



- 종목을 선택하는 기준을 먼저 계산해서 저장 = 과거 9개월 수익률
- 수익률 구하는 방법 = 오늘 종가 / 9개월 전 종가 1.0
- 하지만! 수정주가 이슈 = 이벤트로 인하여 가격이 급격하게 변화 (무상증자, 유상증자, 합병/분할)
- 따라서 수익률을 누적하는 방식이 더 안정적임 (수익률에는 이벤트가 반영되어 있음)

날짜	2016/08/01	2016/08/02	2016/08/03	2016/08/04	2016/08/05	2016/08/06
수익률	1%	2%	-3%	10%	3%	-4%
1+수익률	1.01	1.02	0.97	1.1	1.03	96
	기간 수익률 = 1.01 * 1.02 * 0.97 * 1.1 * 1.03 * 0.96 - 1.0 = 0.23 = 23%					

• 적재 CREATE TABLE 일별시세

```
날짜 DATE PRIMARY,
종목코드 CHAR(7) PRIMARY KEY,
시가 INT,
고가 INT,
저가 INT,
종가 INT,
수익률 DOUBLE,
거래량 INT,
거래대금 INT,
9개월수익률 DOUBLE -- 추가
```

일자-국목 데이터에게 때문에 개준 테이블에 추가

```
CREATE TABLE 영합일테이블
                                            CREATE TABLE 平馬番司오
   MM DATE PRIMARY KEY,
                                                포트폴리오코드 VARCHAR(28) PRIMARY KEY
   전일날짜 DATE,
                                                리탤런상일자 DATE PRIMARY,
   휴장일구분 CHAR(1)
                                               중목코드 CHAR(7) PRIMARY KEY,
   외국휴장일구분 CMAR(1),
                                               비즘 DOUBLE
   요임구분 CHAR(1) NOT NULL,
                                               진입구분 VARCHAR(1) - 0:전일증가, 1:당일시가
   월별마지막양업일구분 CMAR(1),
   주셈마지막양업일구분 CHAR(1)
                                            CREATE TABLE 프로플리오 성과
CREATE TABLE 중국마스터
                                                포트폴리오코드 VARCHAR(28) PRIMARY KEY.
                                               날짜 DATE PRIMARY, KEY,
   중목교드 CHAR(7) PRIMARY KEY,
                                                수의를 DOUBLE
   중독이름 VARCHAR(28)
                                            CREATE TABLE 프로O리오 지표
CREATE TABLE SIMAM
                                                모트폴리오코드 VARCHAR(28) PRIMARY KEY,
   WHY DATE PRIMARY,
                                               초기념짜 DATE,
   총목코드 CHAR(7) PRIMARY KEY,
                                               마지막날짜 DATE.
   APP INT,
                                               연평균수익률 DOUBLE,
   고가 INT.
                                               번통성 DOUBLE
   저가 INT,
                                               会報 DOUBLE.
   증가 INT,
                                               MDD DOUBLE -- Maximum drawdown
   수익률 DOUBLE,
   거라랑 INT,
   거래대급 INT
```

로직 - 포트폴리오 만들기



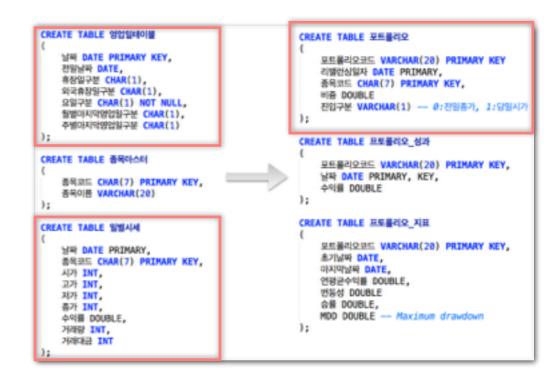
위에서 기확보된 날짜테이블에서 매월 마지막 영업일기준으로 가장 높은 종목을 15개 종목을선택하여 포트폴리오 테이블에 적재

```
# -*- coding: utf-8 -*-
    import conn # DB connection
    import pandas.io.sql as psql
5
6
    def get_target_stock(conn, n):
        sqltext = """
7
8
            SELECT a.날짜, a.종목코드, a.9개월수익률
9
            FROM 일별시세 a, 영업일테이블 b
10
            WHERE a.날짜 = b.날짜
            AND b.매월마지막영업일 = 'Y'
11
12
            ORDER BY a.날짜, b.9개월수익률 desc
        """%(dt)
13
14
        df = psql.read_sql(sqltext, conn)
        df['순위'] = df.groupby('날짜')['9개월수익률'].rank(ascending=False)
15
        return df[df.순위 < n] #각 시점별로 n순위 이내의 종목들 반환
16
17
18
    def insert_target_stock(conn, dt, stock_code, weight):
19
        cur = conn.cursor()
        sqltext = """
20
21
            insert into 포트폴리오(포트폴리오코드, 리밸런싱일자, 종목코드, 비중, 진입시점)
            values('%s', '%s', '%s', %s, '%s')
22
        """%('test', dt, stock_code, weight, '1')
23
24
        cur.execute(sqltext)
25
        conn.commit()
26
        cur.close()
27
28
    df = get_target_stock(conn, 15)
29
    for i in df.index:
30
31
        dt = df.get_value(i, '날짜')
32
        stock_code = df.get_value(i, 'stock_code')
33
        insert target stock(conn, dt, stock code, 1/15)
```

예시) 포트폴리오 = (시점, 종목, 비중) 의 조합

날짜	종목	비중	진입시점
2016/08/0	삼성전자	50%	전일종가
2016/08/0	16/08/0 메리츠증권 50%		전일종가
2016/08/1	삼성전자	33%	전일종가
2016/08/1	현대차	33%	전일종가
2016/08/1	엔씨소프트	33%	전일종가

8/8일에 시작해서 8/10에 주식을 교체하는 포트폴리오



로직 - 성과계산 예시

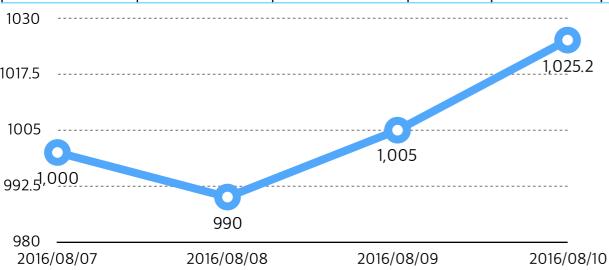


포트폴리오

날짜	종목	비중	진입시점
2016/08/08	삼성전자	50%	전일종가
2016/08/08	메리츠증권	50%	전일종가
2016/08/10	삼성전자	33%	전일종가
2016/08/10	현대차	33%	전일종가
2016/08/10	엔씨소프트	33%	전일종가

포트폴리오 성과

날짜	종목	비중	진입시점	종목수익률	장전	장후	포트폴리오	포트폴리오 수익률
						1000		0%
2016/08/08	삼성전자	50%	전일종가	1%	500	505.0(500*1.01)		
2016/08/08	메리츠증권	50%	전일종가	-3%	500	500 485.0(500*0.97) 990 (505+485) -		-1% (990/1000-1)
2016/08/09	삼성전자			2%	505	515.1(505*1.02)		
2016/08/09	메리츠증권			1%	485	489.9(485*1.01)	1.01) 1005 (515.1+489.9) 1.51%(10	
2016/08/10	삼성전자	33%	전일종가	3%	335 (1005/3)	345.1(335*1.03)	5.1(335*1.03)	
2016/08/10	현대차	33%	전일종가	2%	335	341.7(335*1.02)	12)	
2016/08/10	엔씨소프트	33%	전일종가	1%	335	338.4(335*1.01)	8.4(335*1.01) 1025.2 2%(1025/100	



- · 전략 -> (시점, 종목, 비중) -> 성과
- 수취한 개별 종목의 수익률로 성과 계산
- 8/8일 기준으로 1000원을 투자했을 때를 가정하여 성과 계산
- 포트폴리오 날짜가 아닌 날은 전일자 금액을 그대로 갖고옴
- 거래비용을 고려해야 함
- · 총 수익률 = 1025.2/1000 = 2.52%

로직 - 실제 결과



- " 오른 주식이 더 오르는 것 같은데?"
- " 9개월동안 가장 많이 오른 주식 15개를 똑같은 비중으로 들고가면 어떨까?"
- " 주식은 매달 말에 교체하는 것이 어떨까?

수익률

항목	포트폴리오			
1주	0.56%			
1개월	-3.71%			
3개월	-8.07%			
6개월	18.54%			
1년	8.61%			
3년	117.46%			



http://www.financipe.com/simul/overview?group_id=MMTPRC&code=MMTPRC09M15N

로직 - 성과 분석 (지표)



연환산 기간수익률	여러 포트폴리오를 비교하기 위 해서 1년단위로 수익률 계산
기간별 수익률	현재로부터 1일, 1주일, 3달, 6달 수익률
업종비중	어떤 업종에 치우쳐 있는가?
MDD (Maximum drawdown)	최대 손실폭, 얼마나 위험한 포트 폴리오인가?
턴오버비율	얼마나 자주 종목이 바뀌는 포트 폴리오인가?
변동성	수익률이 얼마나 안정적인가?
승률	내가 돈을 벌 확률은?
TR(Tracking record)	시장지수와 얼마나 다른가?
IR(Information ratio)	알파를 내고 있는가?

	요약	
초기투자금액	2012/01/02	100,000,000
최종계좌잔고	2016/08/05	201,927,868
거래비용		20,776,462
총수익률	2016/08/05	101.93%
BM수익률	2016/08/05	10.55%
리밸런싱 횟수		239 (평균 53.06회/年)
총 영업일		1135
MDD		-34.39%
숭률		53.92%
BM대비승률		26.70%

통계 (연환산)							
항목 포트폴리오 KOSPI							
연평균수익률	16.93%	2.24%					
변동성	29.60%	12.81%					
하방변동성	22.64%	8.71%					
KOSPI대비 초과수익률	13.68%						
TR(Tracking error)	28.94%						
IR(Information ratio)	0.47						
연평균 회전율	1087.30%						
* 모든 통계치는 연률화 되었습니다.							

로직 - 성과 비교



전략코드		전략이름	17	Today	수익률(1W) 🎼	수익률(1M)	수익률(3M)	수익률(6M)	수익률(1Y)
MMTPRC12M10N		모멘텀 탑 10(12개월 수익률)		-0.74%	0.95%	11.46%	11.46%	-1.45%	-4.429
MMTPRC03M10N		모멘텀 탑 10(3개월 수익률)	(X)	-0.20%	0.08%	1.77%	1.77%	-25.22%	-5.21%
MMTPRC06M10N		모멘텀 탑 10(6개월 수익률)	(X)	0.24%	3.39%	10.87%	10.87%	-14.92%	5.59%
MMTPRC09M10N		모멘텀 탑 10(9개월 수익률)		3.11%	6.48%	15.87%	15.87%	21.99%	51.429
MMTPRC12M15N		모멘텀 탑 15(12개월 수익률)	(X)	-0.08%	2.67%	12.32%	12.32%	6.35%	-0.33%
MMTPRC03M15N		모멘텀 탑 15(3개월 수익률)	(X)	0.11%	2.25%	6.22%	6.22%	-17.09%	-5.75
MMTPRC06M15N		모멘텀 탑 15(6개월 수익률)		-0.41%	2.95%	9.74%	9.74%	-29.07%	-7.289
MMTPRC09M15N	-	모멘텀 탑 15(9개월 수익률)	(x)	1.83%	3.76%	8.91%	8.91%	20.37%	41.34
MMTPRC12M20N	Control of the Control	모멘텀 탑 20(12개월 수익률)		0.17%	3.73%	13.21%	13.21%	2.76%	-8.73
MMTPRC03M20N		모멘텀 탑 20(3개월 수익률)		0.15%	1.63%	4.90%	4.90%	-14.91%	-5.56
MMTPRC06M20N		모멘텀 탑 20(6개월 수익률)	(X)	-0.37%	1.52%	7.73%	7.73%	-22.59%	-2.30
MMTPRC09M20N		모멘텀 탑 20(9개월 수익률)		1.11%	2.27%	5.74%	5.74%	6.08%	27.71
MMTPRC12M05N		모멘텀 탑 5(12개월 수익률)	(X)	-1.39%	-1.13%	8.03%	8.03%	-2.56%	25.079
MMTPRC03M05N		모멘텀 탑 5(3개월 수익률)	(X)	-2.20%	-1.38%	2.38%	2.38%	-35.48%	-30.039
MMTPRC06M05N		모멘텀 탑 5(6개월 수익률)		0.15%	3.96%	7.59%	7.59%	-11.41%	-9.739
MMTPRC09M05N		모멘텀 탑 5(9개월 수익률)	(x)	2.82%	3.58%	7.74%	7.74%	12.63%	91.939

우리의 떼시

여러 포트폴리오를 비교 전략에 대한 리스크 분산

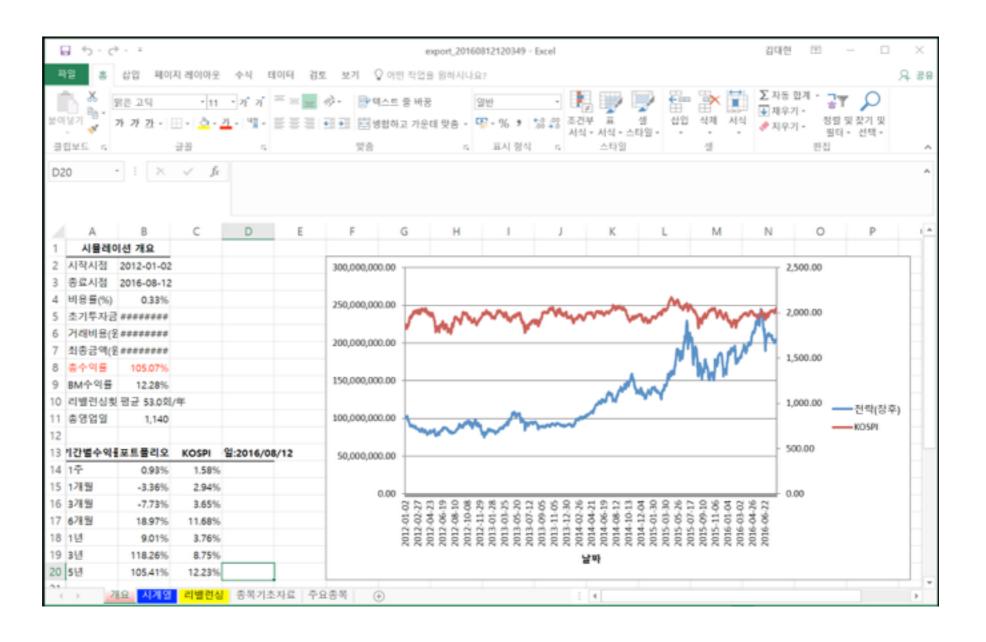
이행 - 레포트 작성



결과물을 여러 형태의 결과물로 반환

- 1) Flask
- 2) Xlsxwriter(**)
- 3) Slack
- 4) 실제 주문

化对性工作 全部区



이슈 - 과적합(Overfitting)



라게 급취기의 원이십시다

- · 숫자를 맹신하지 마세요 **←**
- 과거의 성과가 미래의 수익률을 절대로 보장하지 않습니다
- In sample, Out of sample
- · 전략에 의한 주식투자란 **과거가 되풀이 된다**에 투자하는 것
- · 자신의 아이디어를 증명하는 수준으로 사용하세요 "과전 너 orolfunnt 丈는1+?"
- 데이터 전처리의 중요성은 아무리 말해도 지나치지 않습니다.
- 노이즈 + 노이즈 = 노이즈

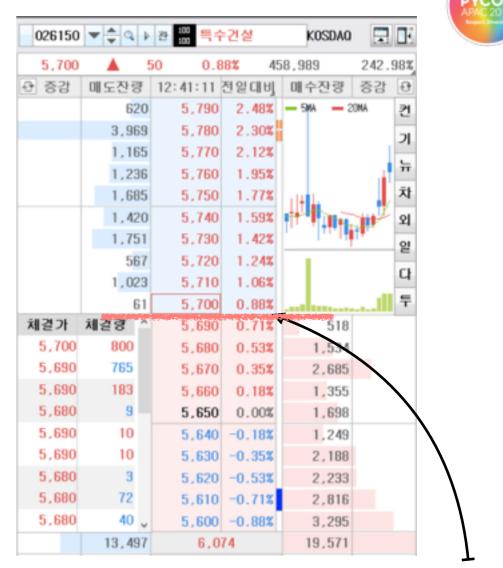
이슈 - 거래비용

• 세금

- · 주식 매도시 세금 (0.3%)
- · 선물/옵션은 매도시 세금 없음 (당가 주네)

· 슬리피지

- 내가 원하는 가격에 들어갈 수가 없습니다
- 시가총액이 작은 종목일 수록 비용이 큽니다



出北 知望这个处程

- · 통상적으로 제세금 0.3% + 슬리피지 0.5% = **0.8%**를 매도할때마다 지불합니다
 - · 총 금액을 매일 사고 팔 경우 한달에 약 16% 손실을 보고 시작합니다

Financipe



http://www.financipe.com/strategytable

정리



python					
아이디어					
데이터					
분석					
확신					
투자					

이런 것들은 파이썬이기에 가능합니다.

- 1인 개발에 맞는 압도적 생산성
- 시스템 트레이딩의 맥가이버 칼
- · 분석에 사용될 수 있는 훌륭한 라이브러리

전략 기반 투자의 장점

- 운에 의존하지 않습니다
- 비이성적인 행위를 최소화 시킬 수 있습니다
- 확신을 가질 수 있으므로 장기투자에 유리합니다
- 일반적인 투자자보다 더 높은 시야를 갖습니다

자신만의 투자 시스템을 만드세요



"수치로 확인하지 않고서는 한 발짝도 움직이지 마라"

서울대 문병로 교수



감사합니다