

5일 이동 평균선이 오늘 종가보다 위에 위치해 있다.

```
import FinanceDataReader as fdr
%matplotlib inline
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import warnings
warnings.filterwarnings('ignore')

pd.options.display.float_format = '{:,.3f}'.format
```

rolling(5) 을 이용하여 이동평균선을 만듭니다. 그리고 당일의 종가보다 크면, 1 아니면 0 인 변수 'flag' 을 생성합니다. 이 가설은 검증이 쉬운 것 같습니다.

```
mdl_data = pd.read_pickle('mdl_data.pkl') # 수익률 결과가 있는 데이터
mdl_data.head().style.set_table_attributes('style="font-size: 12px"')
```

	open	high	low	close	volume	change	code	name	kosdaq_return	return	win_market	close_r1	close_r2	close_r3	close_r4	close_r5	max_close
2021-01-05	2270	2285	2200	2250	410263	-0.004425	060310	3S	1.008326	0.995575	0	1.017778	1.017778	0.997778	0.966667	0.971111	1.017778
2021-01-06	2225	2310	2215	2290	570349	0.017778	060310	3S	0.995567	1.017778	1	1.000000	0.980349	0.949782	0.954148	0.949782	1.000000
2021-01-07	2290	2340	2240	2290	519777	0.000000	060310	3S	1.007612	1.000000	0	0.980349	0.949782	0.954148	0.949782	0.958515	0.980349
2021-01-08	2300	2315	2225	2245	462568	-0.019651	060310	3S	0.998918	0.980349	0	0.968820	0.973274	0.968820	0.977728	0.973274	0.977728
2021-01-11	2230	2275	2130	2175	409057	-0.031180	060310	3S	0.988702	0.968820	0	1.004598	1.000000	1.009195	1.004598	1.002299	1.009195

```
kosdaq_list = pd.read_pickle('kosdaq_list.pkl')

data_h2 = pd.DataFrame()

for code in kosdaq_list['code']:

    data = mdl_data[mdl_data['code']==code].sort_index().copy()

    data['5day_ma'] = data['close'].rolling(5).mean() # 5일 이동평균선
    data['flag'] = (data['close'] < data['5day_ma']).astype(int) # 5일 이동평균선이 종가보다 크면 1, 아니면 0

    data['max_close'] =
data[['close_r1','close_r2','close_r3','close_r4','close_r5']].max(axis=1) # 5 영업일 중
가 수익률 중 최고 값
    data.dropna(subset=
['5day_ma','close_r1','close_r2','close_r3','close_r4','close_r5'], inplace=True) #
missing 이 있는 행은 제거

    data_h2 = pd.concat([data, data_h2], axis=0)

data_h2.to_pickle('data_h2.pkl')
```

'flag' 가 0 인 경우와 1 인 경우를 비교해보니 이 가설은 데이터가 강하게 뒷받침하지 못하고 있습니다.

```
data_h2 = pd.read_pickle('data_h2.pkl')
data_h2.groupby('flag')['max_close'].describe().style.set_table_attributes('style="font-size: 12px"').format(precision=3)
```

	count	mean	std	min	25%	50%	75%	max
flag								
0	209501.000	1.032	0.077	0.326	1.000	1.012	1.044	3.703
1	211348.000	1.034	0.068	0.700	0.999	1.019	1.052	3.139

T-Test 를 해보겠습니다. T-Test 는 두 집단의 평균이 서로 유의미하게 다른 지 확인하는 검정입니다. 귀무가설이 “두 집단의 평균이 같다” 이기 때문에, p -value 가 유의수준(0.01) 보다 작으면 귀무가설을 기각합니다. 결과를 보니 P-Value 가 유의수준(0.01) 보다 큼니다. 따라서 귀무가설을 기각할 수 없습니다. 즉, flag 가 0 인 집단과 1 인 집단간의 차가 유의미하지 않은 것으로 판단됩니다. 왜 각 집단에서 샘플을 200 개만 뽑아서 테스트를 하는 지 궁금한 독자도 있으실 것 같습니다. 통계 검정은 샘플의 수가 많아지면 p value 가 작게 나오는 경향이 있습니다. 그렇게 되면 유의미하게 차이가 없는데도, 서로 다르다고 통계 결과가 나오게 됩니다.

```
from scipy import stats
a = data_h2[data_h2['flag']==0]['max_close'].sample(200)
b = data_h2[data_h2['flag']==1]['max_close'].sample(200)

stats.ttest_ind(a, b, equal_var=False)
```

```
Ttest_indResult(statistic=-1.8358785283648644, pvalue=0.06714153725869931)
```

위 가설은 비교적 증명하기가 쉬웠습니다. 이번에는 5일선과 20일 이동평균선이 만나는 골든크로스에서 매수를 하면 어떤지 보겠습니다. 골든 크로스에서 매수한다고 더 좋은 수익율을 보장하지 않는 것 같습니다.

```
kosdaq_list = pd.read_pickle('kosdaq_list.pkl')

data_h2 = pd.DataFrame()

for code in kosdaq_list['code']:

    data = mdl_data[mdl_data['code']==code].sort_index().copy()
    data['5day_ma'] = data['close'].rolling(5).mean() # 5일 이동평균선
    data['20day_ma'] = data['close'].rolling(20).mean() # 20일 이동평균선
    data['golden_cross'] = (data['5day_ma'].shift(1) < data['20day_ma'].shift(1))*
    (data['5day_ma'] > data['20day_ma']).astype(int) # 5일선이 20일 이동평균선보다 작았다가 커
    지는 시점

    data['max_close'] =
    data[['close_r1','close_r2','close_r3','close_r4','close_r5']].max(axis=1) # 5 영업일 중
    가 수익율 중 최고 값
    data.dropna(subset=
    ['5day_ma','20day_ma','golden_cross','close_r1','close_r2','close_r3','close_r4','close_
    r5'], inplace=True) # missing 이 있는 행은 제거

    data_h2 = pd.concat([data, data_h2], axis=0)

data_h2.to_pickle('data_h2.pkl')
```

```
data_h2 = pd.read_pickle('data_h2.pkl')
data_h2.groupby('golden_cross')
['max_close'].describe().style.set_table_attributes('style="font-size:
12px"').format(precision=3)
```

	count	mean	std	min	25%	50%	75%	max
golden_cross								
0	387430.000	1.032	0.071	0.326	1.000	1.016	1.048	3.703
1	12164.000	1.033	0.079	0.700	0.997	1.015	1.047	2.852