마켓 데이터 분석: 이동평균과 시그널

이승준 fb.com/plusjune

pandas 이동평균

- pandas.stats.rolling_mean()

```
df['MA_5'] = pd.stats.moments.rolling_mean(df['Adj Close'], 5)
df['MA_20'] = pd.stats.moments.rolling_mean(df['Adj Close'], 20)
df['diff'] = df['MA_5'] - df['MA_20']
df.head(10)
```

	Open	High	Low	Close	Volume	Adj Close	MA_5	MA_20	diff
Date									
2013-01-01	1522000	1522000	1522000	1522000	0	1509061.31	NaN	NaN	NaN
2013-01-02	1533000	1576000	1527000	1576000	228900	1562602.25	NaN	NaN	NaN
2013-01-03	1582000	1584000	1543000	1543000	284500	1529882.78	NaN	NaN	NaN
2013-01-04	1540000	1542000	1510000	1525000	259900	1512035.80	NaN	NaN	NaN
2013-01-07	1515000	1528000	1500000	1520000	252200	1507078.31	1524132.090	NaN	NaN
2013-01-08	1513000	1517000	1498000	1500000	276400	1487248.33	1519769.494	NaN	NaN
2013-01-09	1500000	1513000	1491000	1500000	253100	1487248.33	1504698.710	NaN	NaN
2013-01-10	1515000	1534000	1500000	1530000	293200	1516993.30	1502120.814	NaN	NaN
2013-01-11	1548000	1548000	1507000	1533000	238200	1519967.79	1503707.212	NaN	NaN
2013-01-14	1539000	1552000	1528000	1552000	159900	1538806.27	1510052.804	NaN	NaN

골든크로스, 데드크로스

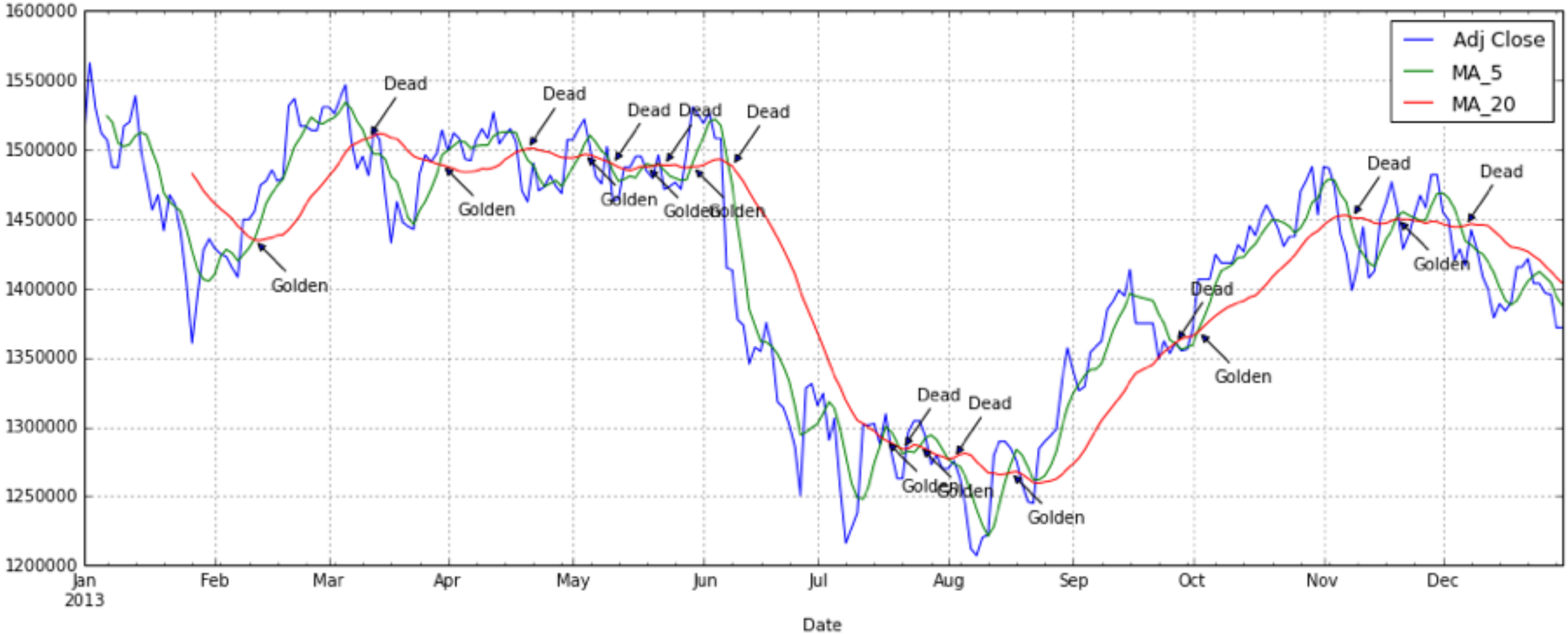
단기와 장기 이동평균의 차이값 (MA_5 - MA_20)를 비교

크로스: 차이값 x 이전 차이값 < 0 (즉, 이전 값과 부호가 바뀌는 경우)

```
prev_key = prev_val = 0

  for key, val in df['diff'][1:].iteritems():
      if val == 0:
          continue
      if val * prev_val < 0 and val > prev_val:
          print '[golden]', key, val
      if val * prev_val < 0 and val < prev_val:
          print '[dead]', key, val
      prev_key, prev_val = key, val
```

```
[golden] 2013-02-12 00:00:00 842.774000001
[dead] 2013-03-12 00:00:00 -4957.4955
[golden] 2013-03-29 00:00:00 7237.9425
[dead] 2013-04-19 00:00:00 -7882.416
[golden] 2013-05-03 00:00:00 7287.5155
... ...
```

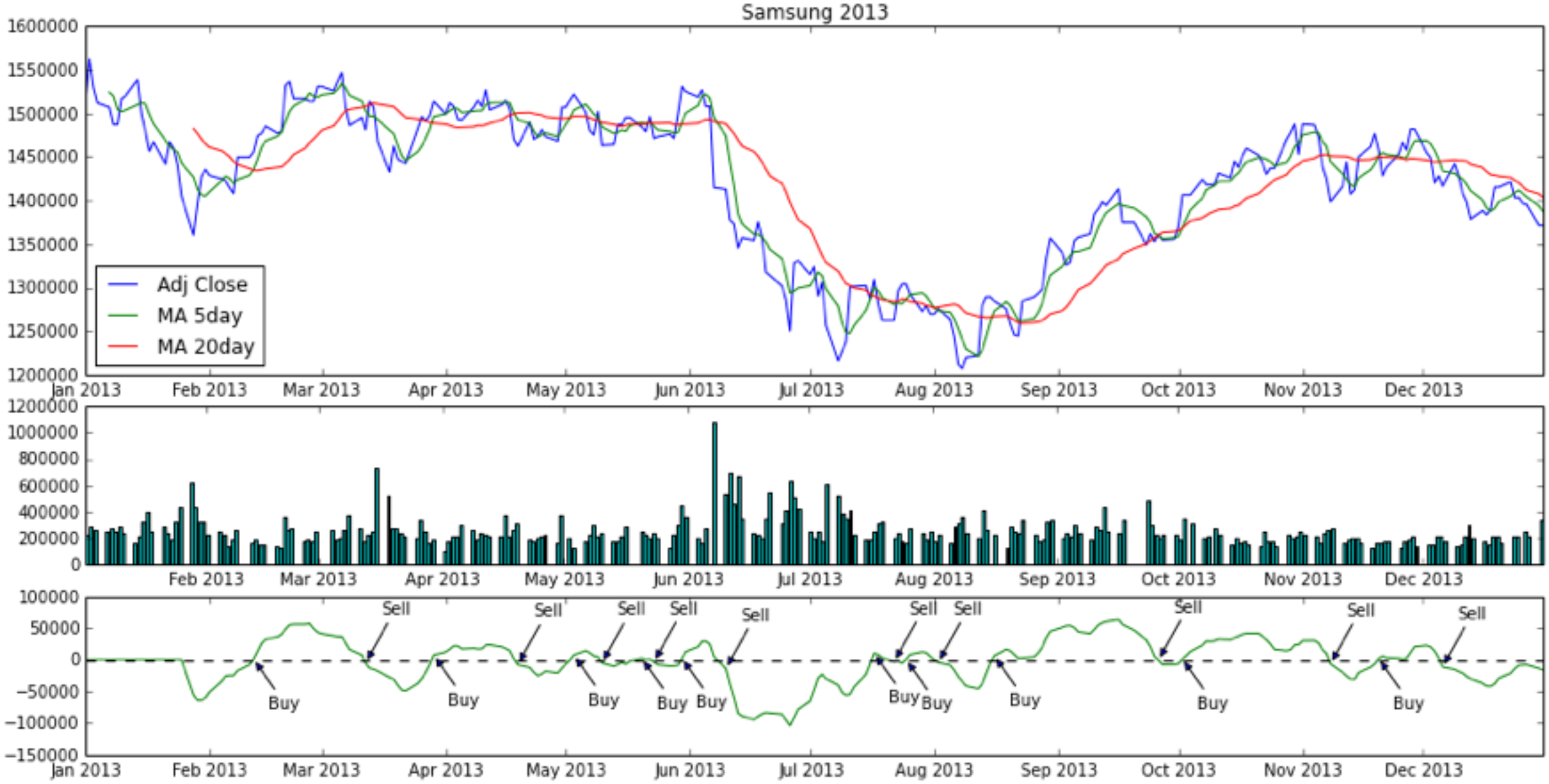
```
ax = df[['Adj Close', 'MA_5', 'MA_20']].plot(figsize(16,6))
  prev_key = prev_val = 0

  for key, val in df['diff'][1:].iteritems():
      if val == 0:
          continue

      if val * prev_val < 0 and val > prev_val:
          ax.annotate('Golden', xy=(key, df['MA_20'][key]), xytext=(10,-30),
                      textcoords='offset points', arrowprops=dict(arrowstyle='-|>'))
      if val * prev_val < 0 and val < prev_val:
          ax.annotate('Dead', xy=(key, df['MA_20'][key]), xytext=(10,30),
                      textcoords='offset points', arrowprops=dict(arrowstyle='-|>'))

      prev_key, prev_val = key, val
```


Adj Close
MA_5
MA_20
Dead
Golden
Date
Jan
2013
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
1600000
1550000
1500000
1450000
1400000
1350000
1300000
1250000
1200000

```
fig = matplotlib.pyplot.gcf()
  fig.set_size_inches(16, 8)

  # price (가격)
  price_chart = plt.subplot2grid((4,1), (0, 0), rowspan=2)
  price_chart.plot(df.index, df['Adj Close'], label='Adj Close')
  price_chart.plot(df.index, df['MA_5'], label='MA 5day')
  price_chart.plot(df.index, df['MA_20'], label='MA 20day')

  plt.title(u'Samsung 2013')
  plt.legend(loc='best')

  # volume (거래량)
  vol_chart = plt.subplot2grid((4,1), (2,0), rowspan=1)
  vol_chart.bar(df.index, df['Volume'], color='c')
```

- # 이동평균의 차이

```
signal_chart = plt.subplot2grid((4,1), (3,0), rowspan=1)
signal_chart.plot(df.index, df['diff'].fillna(0), color='g')
plt.axhline(y=0, linestyle='--', color='k')

prev_key = prev_val = 0 # sell, buy annotate

for key, val in df['diff'][1:].iteritems():
    if val == 0:
        continue
    if val * prev_val < 0 and val > prev_val:
        signal_chart.annotate('Buy', xy=(key, df['diff'][key]), xytext=(10,-30),
textcoords='offset points', arrowprops=dict(arrowstyle='-|>'))
    if val * prev_val < 0 and val < prev_val:
        signal_chart.annotate('Sell', xy=(key, df['diff'][key]), xytext=(10,30),
textcoords='offset points', arrowprops=dict(arrowstyle='-|>'))
    prev_key, prev_val = key, val
```


Samsung 2013
Adj Close
MA 5day
MA 20day
Buy
Sell
Jan 2013
Feb 2013
Mar 2013
Apr 2013
May 2013
Jun 2013
Jul 2013
Aug 2013
Sep 2013
Oct 2013
Nov 2013
Dec 2013