

## 大阳线和大阴线

目前，我们讨论日 K 线，尤其是耳熟能详的两个概念就是阳线和阴线。很多人可能主要通过日 K 线的图来判别这两种 K 线。例如，一般中部的实体以空白或红色表示为阳线，中部的实体以黑色或绿色的为阴线。那么，如何量化判断和应用这两个概念呢？计算机如何识别呢？

首先，我们要从基本概念入手。阳线是证券市场上指收盘价高于开盘价的 K 线。阴线是收盘价低于开盘价的 K 线。这样一来，量化是不是只需要证券的日 K 线（开，高，低，收）就可以判断当日的 K 线是属于阳线还是阴线？阳线和阴线的用法太灵活了，有太多的判断技术。一般的，阳线意味买方气势旺盛，力量强大，后市看涨；阴线意味卖方气势旺盛，后市看跌。而且，阳线和阴线的力度都是跟 K 线柱的长度成正比。用量化的语言表示：阳线和阴线的力度和日 K 线中开盘与收盘价格差的绝对值成正比。常见的判断规则如下：

### 1. 阳线

- 1.1 若证券股价涨势初期出现大阳线，代表股价很可能继续上涨；
- 1.2 若证券股价连续上涨后出现大阳线，代表有可能因多方力量过分消耗导致股价见顶回落；
- 1.3 若证券股价连续下跌后出现大阳线，代表股价可能见底回升。

### 2. 阴线

- 2.1 若大阴线出现在证券股价大涨之后，则后市很有可能下跌；
- 2.2 若大阴线出现在证券股价上升初中期，则考虑主力庄家在利用利空消息进行震仓洗盘，只有股价不跌破重要支撑位，仍可持股；
- 2.3 若大阴线出现在下跌途中，则按时股价还有下跌空间；
- 2.4 若大阴线出现在较大跌幅之后，则暗示空头能力释放的差不多，股价可能止跌回升。

接下来，我将通过 python 程序调用 [baostock](#) (baostock 是免费证券数据的 python 接口，具体信息参考：[www.baostock.com](#)) 对大阳线和大阴线这两个指标进行量化。同时，针对 1.3 和 2.1 这两种情况，进行统计分析来验证这两种判断方法的准确度。

具体代码如下：

```
#!/user/bin/env python
# -*- coding:utf-8 -*-

"""
利用 Baostock 的数据源，在给定起止时间内计算单个证券在这段时间的阳线和阴线指标。
"""

import baostock as bs
import pandas as pd

def judge_kline_category(code, startdate, enddate):
    """判断证券在起止时间内的每日 K 线类别：阳线，阴线。
```

```
:param code: 证券代码
:param startdate: 起始日期
:param enddate: 截止日期
:return:
"""
login_result = bs.login(user_id='anonymous', password='123456')
print(login_result.error_msg)

# 获取股票日 K 线数据,adjustflag      复权状态(1: 后复权, 2: 前复权,
3: 不复权)
rs = bs.query_history_k_data(code,

"date,code,open,high,low,close,tradeStatus",
                                start_date=startdate,
end_date=enddate,
                                frequency="d", adjustflag="3")

# 打印结果集
result_list = []
while (rs.error_code == '0') & rs.next():
    # 获取一条记录, 将记录合并在一起
    result_list.append(rs.get_row_data())
df_init = pd.DataFrame(result_list, columns=rs.fields)
# 剔除停牌数据
df_status = df_init[df_init['tradeStatus'] == '1']
df_status['open'] = df_status['open'].astype(float)
df_status['high'] = df_status['high'].astype(float)
df_status['low'] = df_status['low'].astype(float)
df_status['close'] = df_status['close'].astype(float)
df_status['kline_category'] = df_status.apply(
    lambda x: judge_function(x.open, x.close), axis=1)
df_status.to_csv('D:/df.csv')
return df_status

def kline_application(df, N):
    """已知证券在起止时间内的每日 K 线类别: 阳线, 阴线。
    做如下统计:
    情景 1. 若证券股价连续 N 天下跌后出现大阳线, 次日股价开盘上涨的次数
    (第 N+2 天)
    情景 2. 若证券股价连续 N 天上涨之后出现大阴线, 次日股价开盘下跌的次数
    (第 N+2 天)

    :return:
```

```
"""
    daycounts = df.shape[0]
    df['kline_num'] = [1 if x == 'positive' else 0 for x in
df['kline_category']]
    df['scene'] = 0
    total_counts_1 = 0 # 计算情景 1 中证券股价连续 N 天下跌后出现大阳线的次
数
    total_counts_1_sub = 0 # 计算情景 1 出现的次数
    total_counts_2 = 0 # 计算情景 2 中证券证券股价连续 N 天上涨之后出现大阴
线的次数
    total_counts_2_sub = 0 # 计算情景 2 出现的次数
    for i in range(0, daycounts - N - 1):
        kline_num_counts = 0
        for j in range(0, N):
            kline_num_counts += df.iloc[i + j, 8]
        if kline_num_counts == N and df.iloc[i + N, 8] == 0:
            # 表明该证券连续 N 天上涨后出现了大阴线
            total_counts_2 += 1
            if df.iloc[i + N + 1, 2] < df.iloc[i + N, 5]:
                total_counts_2_sub += 1
                df.iloc[i + N, 9] = 2 # 表明这一天属于情景 2
            if kline_num_counts == 0 and df.iloc[i + N, 8] == 1:
                # 表明该证券连续 N 天下跌后出现了大阳线
                total_counts_1 += 1
                if df.iloc[i + N + 1, 2] > df.iloc[i + N, 5]:
                    total_counts_1_sub += 1
                    df.iloc[i + N, 9] = 1 # 表明这一天属于情景 1
    df.to_csv('D:/df2.csv')
    print("证券代码: " + df['code'][0])
    print("证券股价连续 N 天下跌后出现大阳线的总次数: " +
str(total_counts_1))
    print("若证券股价连续 N 天下跌后出现大阳线, 次日股价开盘上涨的次数 (第
N+2 天):" + str(total_counts_1_sub) + ", 占比: "
        + str(total_counts_1_sub / total_counts_1))
    print("证券股价连续 N 天上涨之后出现大阴线的总次数: " +
str(total_counts_2))
    print("若证券股价连续 N 天上涨之后出现大阴线, 次日股价开盘下跌的次数 (第
N+2 天):" + str(total_counts_2_sub) + ", 占比: "
        + str(total_counts_2_sub / total_counts_2))
    return(total_counts_1, total_counts_1_sub, total_counts_2,
total_counts_2_sub)

def judge_function(open, close):
```

```

    if open > close:
        return 'negative'
    else:
        return 'positive'

if __name__ == '__main__':
    code = "sh.600000"
    startdate = "2000-01-01"
    enddate = "2018-01-01"
    N = 3
    df = judge_kline_category(code, startdate, enddate)
    kline_application(df, N)

```

输出结果如下（红色是警告，可以忽略）：

```

D:\software\Anaconda3\python.exe D:/software/eclipse_workspace/QuantitativeTrading/Strategies/technical_indicators/positiveline_negativeline.py
login success!
success
D:/software/eclipse_workspace/QuantitativeTrading/Strategies/technical_indicators/positiveline_negativeline.py:50: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy
  df_status['open'] = df_status['open'].astype(float)
D:/software/eclipse_workspace/QuantitativeTrading/Strategies/technical_indicators/positiveline_negativeline.py:51: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy
  df_status['high'] = df_status['high'].astype(float)
D:/software/eclipse_workspace/QuantitativeTrading/Strategies/technical_indicators/positiveline_negativeline.py:52: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy
  df_status['low'] = df_status['low'].astype(float)
D:/software/eclipse_workspace/QuantitativeTrading/Strategies/technical_indicators/positiveline_negativeline.py:53: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy
  df_status['close'] = df_status['close'].astype(float)
D:/software/eclipse_workspace/QuantitativeTrading/Strategies/technical_indicators/positiveline_negativeline.py:54: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy
  df_status['kline_category'] = df_status.apply(lambda x: judge_function(x.open, x.close), axis = 1)
证券代码: sh.600000
证券股价连续N天下跌后出现太阳线的总次数: 73
若证券股价连续N天下跌后出现太阳线, 次日股价开盘上涨的次数 (第N+2天) :31,占比: 0.4246575342465753
证券股价连续N天上涨之后出现大阴线的总次数: 100
若证券股价连续N天上涨之后出现大阴线, 次日股价开盘下跌的次数 (第N+2天) :53,占比: 0.53
Process finished with exit code 0

```

从结果上来看，针对浦发银行，情景 2（即 2.1 判别规则）出现的概率有 52%，情景 1（即 1.3 判别规则）出现的概率有 42%。若想更严谨些，我们可以循环所有的证券，求各个判别规则的概率。这些事情就交给读者完成了。有结果请告知我。

df.csv 文件内容如下：

	A	B	C	D	E	F	G	H	I	J
1		date	code	open	high	low	close	tradeStatu	kline_category	
2	0	2012/1/4	sh.6000000	8.54	8.56	8.39	8.41	1	negative	
3	1	2012/1/5	sh.6000000	8.47	8.82	8.47	8.65	1	positive	
4	2	2012/1/6	sh.6000000	8.63	8.78	8.62	8.71	1	positive	
5	3	2012/1/9	sh.6000000	8.72	8.99	8.68	8.95	1	positive	
6	4	2012/1/10	sh.6000000	8.95	9.1	8.88	9.07	1	positive	
7	5	2012/1/11	sh.6000000	9.05	9.1	8.98	9	1	negative	
8	6	2012/1/12	sh.6000000	9	9.21	8.99	9.09	1	positive	
9	7	2012/1/13	sh.6000000	9.11	9.18	8.97	9.04	1	negative	
10	8	2012/1/16	sh.6000000	9	9.12	8.96	8.97	1	negative	
11	9	2012/1/17	sh.6000000	9	9.26	8.79	9.2	1	positive	
12	10	2012/1/18	sh.6000000	9.16	9.23	9	9.04	1	negative	
13	11	2012/1/19	sh.6000000	9.05	9.22	9.02	9.2	1	positive	
14	12	2012/1/20	sh.6000000	9.25	9.43	9.17	9.42	1	positive	
15	13	2012/1/30	sh.6000000	9.4	9.41	9.2	9.22	1	negative	
16	14	2012/1/31	sh.6000000	9.24	9.28	9.16	9.22	1	negative	
17	15	2012/2/1	sh.6000000	9.2	9.26	9.05	9.08	1	negative	
18	16	2012/2/2	sh.6000000	9.12	9.42	9.08	9.4	1	positive	
19	17	2012/2/3	sh.6000000	9.35	9.5	9.32	9.44	1	positive	
20	18	2012/2/6	sh.6000000	9.46	9.51	9.29	9.39	1	negative	
21	19	2012/2/7	sh.6000000	9.32	9.36	9.18	9.24	1	negative	
22	20	2012/2/8	sh.6000000	9.23	9.48	9.23	9.45	1	positive	
23	21	2012/2/9	sh.6000000	9.45	9.48	9.36	9.4	1	negative	
24	22	2012/2/10	sh.6000000	9.36	9.48	9.32	9.37	1	positive	
25	23	2012/2/13	sh.6000000	9.25	9.37	9.18	9.27	1	positive	

df2.csv 文件内容如下:

	A	B	C	D	E	F	G	H	I	J	K	L
1		date	code	open	high	low	close	tradeStatu	kline_category	kline_num	scene	
2	0	2012/1/4	sh.6000000	8.54	8.56	8.39	8.41	1	negative	0	0	
3	1	2012/1/5	sh.6000000	8.47	8.82	8.47	8.65	1	positive	1	0	
4	2	2012/1/6	sh.6000000	8.63	8.78	8.62	8.71	1	positive	1	0	
5	3	2012/1/9	sh.6000000	8.72	8.99	8.68	8.95	1	positive	1	0	
6	4	2012/1/10	sh.6000000	8.95	9.1	8.88	9.07	1	positive	1	0	
7	5	2012/1/11	sh.6000000	9.05	9.1	8.98	9	1	negative	0	0	
8	6	2012/1/12	sh.6000000	9	9.21	8.99	9.09	1	positive	1	0	
9	7	2012/1/13	sh.6000000	9.11	9.18	8.97	9.04	1	negative	0	0	
10	8	2012/1/16	sh.6000000	9	9.12	8.96	8.97	1	negative	0	0	
11	9	2012/1/17	sh.6000000	9	9.26	8.79	9.2	1	positive	1	0	
12	10	2012/1/18	sh.6000000	9.16	9.23	9	9.04	1	negative	0	0	
13	11	2012/1/19	sh.6000000	9.05	9.22	9.02	9.2	1	positive	1	0	
14	12	2012/1/20	sh.6000000	9.25	9.43	9.17	9.42	1	positive	1	0	
15	13	2012/1/30	sh.6000000	9.4	9.41	9.2	9.22	1	negative	0	0	
16	14	2012/1/31	sh.6000000	9.24	9.28	9.16	9.22	1	negative	0	0	
17	15	2012/2/1	sh.6000000	9.2	9.26	9.05	9.08	1	negative	0	0	
18	16	2012/2/2	sh.6000000	9.12	9.42	9.08	9.4	1	positive	1	0	
19	17	2012/2/3	sh.6000000	9.35	9.5	9.32	9.44	1	positive	1	0	
20	18	2012/2/6	sh.6000000	9.46	9.51	9.29	9.39	1	negative	0	0	
21	19	2012/2/7	sh.6000000	9.32	9.36	9.18	9.24	1	negative	0	0	
22	20	2012/2/8	sh.6000000	9.23	9.48	9.23	9.45	1	positive	1	0	
23	21	2012/2/9	sh.6000000	9.45	9.48	9.36	9.4	1	negative	0	0	
24	22	2012/2/10	sh.6000000	9.36	9.48	9.32	9.37	1	positive	1	0	
25	23	2012/2/13	sh.6000000	9.25	9.37	9.18	9.27	1	positive	1	0	