Python 实现 RSI 指标的超买和超卖信息提示

RSI ，即相对强弱指标，是由韦尔斯.怀尔德(Welles Wilder)提出的，是衡量证券自身内在相对强度的指标。相对强弱指数 RSI 是根据一定时期内上涨和下跌幅度之和的比率制作出的一种技术曲线，能够反映出市场在一定时期内的景气程度。因为投资的一般原理认为，投资者的买卖行为是各种因素综合结果的反映，行情的变化最终取决于供求关系，而 RSI 指标正是根据供求平衡的原理，通过测量某一个期间内股价上涨总幅度占股价变化总幅度平均值的百分比，来评估多空力量的强弱程度，进而提示具体操作的。

RSI 公式不仅能够提供这种平滑特征，而且可以产生一个能够在 0-100 之间固定区域变动的指标。怀尔德推荐的默认时间跨度是 14 天，他论证了应用月周期 28 日的一半是有效的。

计算公式：
N 日 RSI =N 日内收盘涨幅的平均值/(N 日内收盘涨幅均值+N 日内收盘跌幅均值) ×100

由上面算式可知 RSI 指标的技术含义，即以向上的力量与向下的力量进行比较，若向上的力量较大，则计算出来的指标上升；若向下的力量较大，则指标下降，由此测算出市场走势的强弱。

市场上一般的规则：（快速 RSI 指 14 日的 RSI，慢速 RSI 指 6 日的 RSI）
1. RSI 金叉：快速 RSI 从下往上突破慢速 RSI 时,认为是买进机会。
2. RSI 死叉：快速 RSI 从上往下跌破慢速 RSI 时,认为是卖出机会
3. 慢速 RSI<20 为超卖状态,为买进机会。
4. 慢速 RSI>80 为超买状态,为卖出机会。

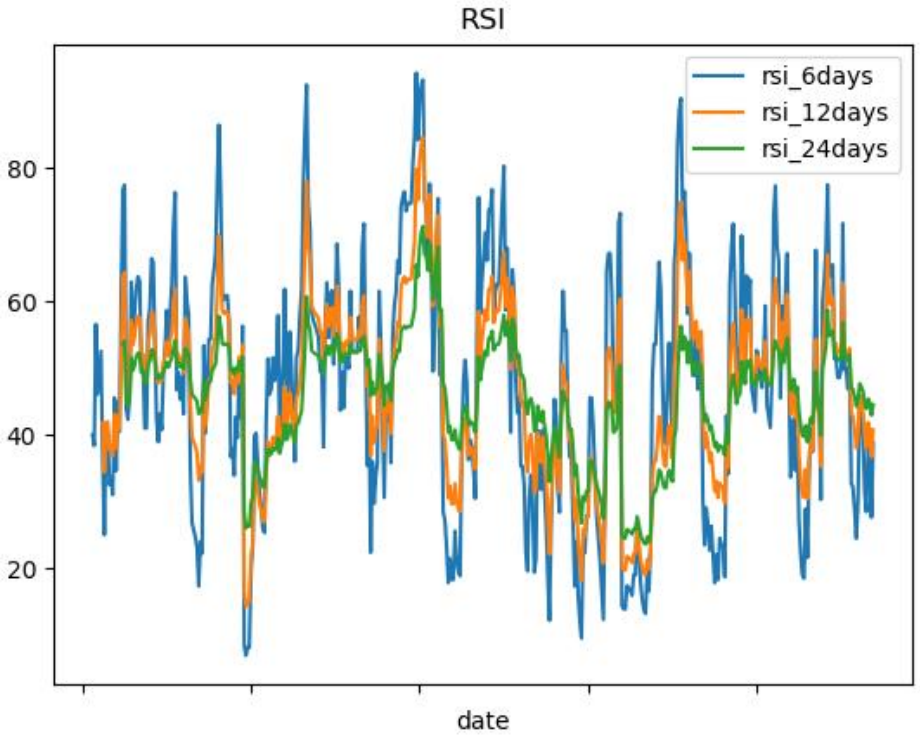
接下来，我将通过 python 程序调用 baostock（baostock 是免费证券数据的 python 接口，具体信息参考：www.baostock.com）实现 RSI 计算，RSI 超卖和超买提示的功能。
具体代码如下：

```
import baostock as bs
import pandas as pd
import talib as ta
import matplotlib.pyplot as plt


def computeRSI(code, startdate, enddate):
    """计算证券在起止时间内的RSI 指标。

        :param code:证券代码
        :param startdate:起始日期
        :param enddate:截止日期
        :return:
    """
```

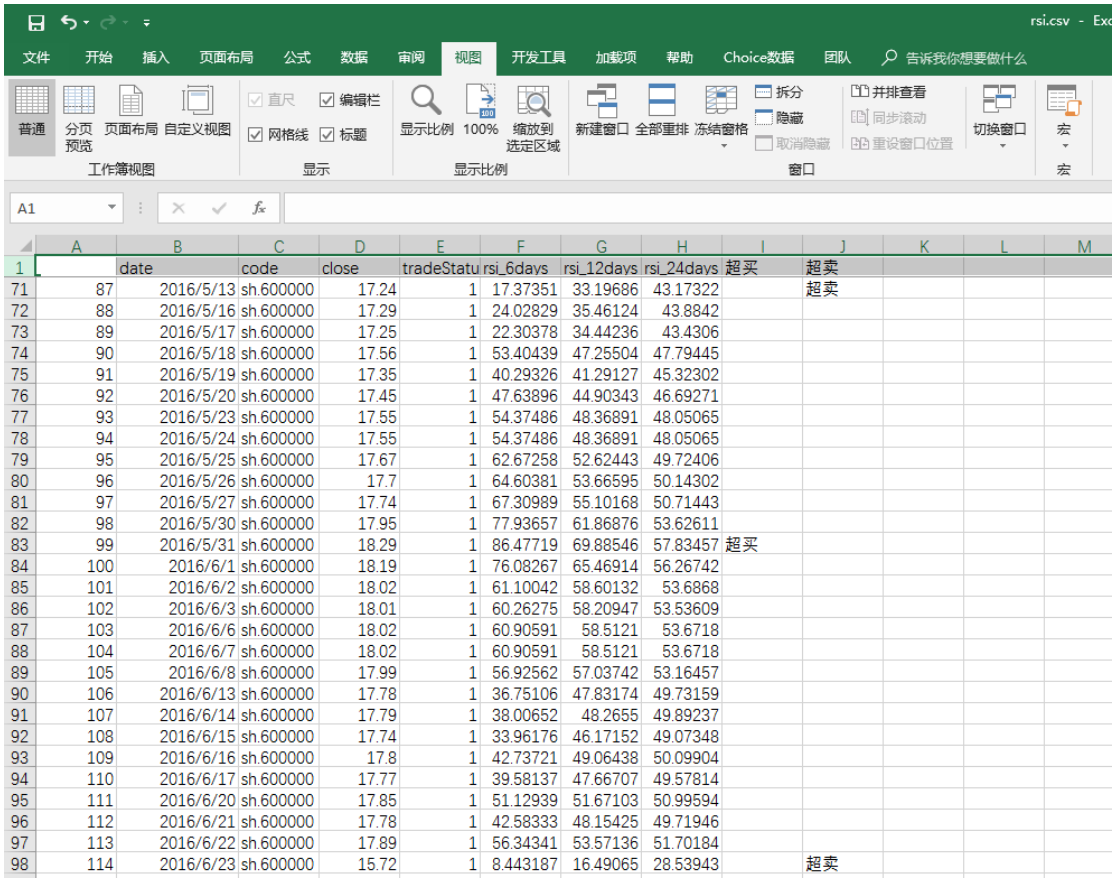
```
    login_result = bs.login(user_id='anonymous', password='123456')
    print(login_result.error_msg)

    # 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,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)
    # 剔除停盘数据
    df_status = df_init[df_init['tradeStatus'] == '1']

    df_status['close'] = df_status['close'].astype(float)

    rsi_12days = ta.RSI(df_status['close'],timeperiod=12)
    rsi_6days = ta.RSI(df_status['close'],timeperiod=6)
    rsi_24days = ta.RSI(df_status['close'],timeperiod=24)
    df_status['rsi_6days'] = rsi_6days
    df_status['rsi_12days'] = rsi_12days
    df_status['rsi_24days'] = rsi_24days

    # RSI 超卖和超买
    rsi_buy_position = df_status['rsi_6days'] > 80
    rsi_sell_position = df_status['rsi_6days'] < 20
    df_status.loc[rsi_buy_position[(rsi_buy_position == True) & (rsi_buy_position.shift() == False)].index, '超买'] = '超买'
    df_status.loc[rsi_sell_position[(rsi_sell_position == True) & (rsi_sell_position.shift() == False)].index, '超卖'] = '超卖'

    return df_status


if __name__ == '__main__':
    code = "sh.600000"
    startdate = "2016-01-01"
    enddate = "2018-01-01"
    df = computeRSI(code, startdate, enddate)
    df2 = df[['date','rsi_6days','rsi_12days','rsi_24days']]
```

```
    df2.index = df['date']
    df2.plot(title='RSI')
    plt.show()
    df.to_csv("D:\\rsi.csv",encoding='gbk')
```

结果画图如下：



生成的 CSV 文件内容如下：



	A	B	C	D	E	F	G	H	I	J
1		date	code	close	tradeStatu	rsi_6days	rsi_12days	rsi_24days	超买	超卖
71	87	2016/5/13	sh.600000	17.24	1	17.37351	33.19686	43.17322		超卖
72	88	2016/5/16	sh.600000	17.29	1	24.02829	35.46124	43.8842		
73	89	2016/5/17	sh.600000	17.25	1	22.30378	34.44236	43.4306		
74	90	2016/5/18	sh.600000	17.56	1	53.40439	47.25504	47.79445		
75	91	2016/5/19	sh.600000	17.35	1	40.29326	41.29127	45.32302		
76	92	2016/5/20	sh.600000	17.45	1	47.63896	44.90343	46.69271		
77	93	2016/5/23	sh.600000	17.55	1	54.37486	48.36891	48.05065		
78	94	2016/5/24	sh.600000	17.55	1	54.37486	48.36891	48.05065		
79	95	2016/5/25	sh.600000	17.67	1	62.67258	52.62443	49.72406		
80	96	2016/5/26	sh.600000	17.7	1	64.60381	53.66595	50.14302		
81	97	2016/5/27	sh.600000	17.74	1	67.30989	55.10168	50.71443		
82	98	2016/5/30	sh.600000	17.95	1	77.93657	61.86876	53.62611		
83	99	2016/5/31	sh.600000	18.29	1	86.47719	69.88546	57.83457	超买	
84	100	2016/6/1	sh.600000	18.19	1	76.08267	65.46914	56.26742		
85	101	2016/6/2	sh.600000	18.02	1	61.10042	58.60132	53.6868		
86	102	2016/6/3	sh.600000	18.01	1	60.26275	58.20947	53.53609		
87	103	2016/6/6	sh.600000	18.02	1	60.90591	58.5121	53.6718		
88	104	2016/6/7	sh.600000	18.02	1	60.90591	58.5121	53.6718		
89	105	2016/6/8	sh.600000	17.99	1	56.92562	57.03742	53.16457		
90	106	2016/6/13	sh.600000	17.78	1	36.75106	47.83174	49.73159		
91	107	2016/6/14	sh.600000	17.79	1	38.00652	48.2655	49.89237		
92	108	2016/6/15	sh.600000	17.74	1	33.96176	46.17152	49.07348		
93	109	2016/6/16	sh.600000	17.8	1	42.73721	49.06438	50.09904		
94	110	2016/6/17	sh.600000	17.77	1	39.58137	47.66707	49.57814		
95	111	2016/6/20	sh.600000	17.85	1	51.12939	51.67103	50.99594		
96	112	2016/6/21	sh.600000	17.78	1	42.58333	48.15425	49.71946		
97	113	2016/6/22	sh.600000	17.89	1	56.34341	53.57136	51.70184		
98	114	2016/6/23	sh.600000	15.72	1	8.443187	16.49065	28.53943		超卖

至于 RSI 的金叉和死叉的实现，我在之前 KDJ 里有类似的方法，请大家自己动手。现在我们已经实现了 MACD、KDJ、RSI 这三个技术指标。可能有部分人认为计算的结果和交易软件给出的不一样。首先，交易软件给出的是从上市以来的计算结果；其次，我也对比了好几个，发现我计算的结果和新浪财经，腾讯财经提供的指标值比较接近。若还有什么问题，欢迎提问啊。

参考文献：

https://baijiahao.baidu.com/s?id=1571691914816607&wfr=spider&for=pc
https://blog.csdn.net/MARY197011111/article/details/79622184
https://blog.csdn.net/ialexanderi/article/details/75395211
http://bbs.pinggu.org/thread-5763091-1-1.html
Python 包 talib 下载：https://www.lfd.uci.edu/~gohlke/pythonlibs/