

基础不简单之市现率

之前的文章讲解了市盈率，市净率，有没有发现刚好对应了公司三大报表之利润表，资产负债表。那现金流表有没有对应的量化指标呢？当然有啦，这个指标就是市现率。

市现率 (PCF)，指的是股票价格与每股现金流量的比率，即：每股股价 / 每股现金流量。

现金流量是指企业在一定会计期间按照现金收付实现制，通过一定经济活动(包括经营活动、投资活动、筹资活动和非经常性项目)而产生的现金流入、现金流出及其总量情况的总称，即企业一定时期的现金和现金等价物的流入和流出的数量。可以说，现金流量是确保企业的生存与发展、提高企业市场竞争力的重要保障。现金流量如此重要，那么市现率是否也是重要指标呢？答案是肯定的。

根据现金流量的特性，市现率有如下优点：**1. 可靠性高。**自由现金流真实反应了企业的资金动态流动状况，是比较难以操纵的项目；**2. 灵敏度高。**自由现金流及时地反应了企业的经营状况，比损益更为敏感；**3. 真实度高。**现金流量是在不影响企业正常运营和发展的情况下，能够分配给债权人和债务人最大的现金额。从投资者角度来说，这部分金额就是企业为投资者创造的实实在在的价值增值。

以上分析表明市现率可用于评价股票的价格水平和风险水平。市现率越小，表明上市公司的每股现金增加额越多，经营压力越小。相反的，市现率越大，表明公司的经营压力大，投资风险也大。不要忘了，乐视网就是现金流问题而遭遇崩盘的。同 **PE(TTM)**类似，市现率也存在一个滚动的市现率指标，即 **PCF(TTM)**，这里就不作过多解释。

我采用 [BaoStock](#) 的 python 数据接口，查询了某一天的 A 股市场全部证券的 **pcfTTM**。发现 3276 只证券中有 1690 只证券的 **PCFTTM** 小于 0。为何会小于 0？造成这种结果的原因在于分母，分母是公司季报的净现金流，请注意净现金流包括投资现金流、融资现金流和经营现金流三个部分，一般来说，经营现金流为正值（表明企业是营利的），但投资现金流和融资现金流的正负用于判断企业的运营状况比较复杂，不是可简单下结论的，例如，企业加大投资，用现金购买资产，则投资现金流为负，但可能意味着投资的资产未来有较好的收益，卖出资产，缩减投资，则投资现金流为正；融资现金流的情况也是如此，如果偿还了债务，则融资现金流为负，反之，发行债券和股票，则融资现金流为正。三个部分加总的结果即为净现金流，大小和符号也就确定了。

因此，市现率正和负，以及绝对值的大小仅能提供一定的参考，并不能作为公司未来好坏的绝对评价标准。讲解了这么多，我还是贴上一段类似之前 2 篇文章的代码，告诉大家如何查询证券的市现率。

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

import pandas as pd
import baostock as bs
```

```
# 登陆系统
lg = bs.login()
# 显示登陆返回信息
print('login respond error_code:' + lg.error_code)
print('login respond error_msg:' + lg.error_msg)

# 获取某一天的全市场的证券和指数代码
rs = bs.query_all_stock(day="2018-06-28")
print('query_all_stock respond error_code:' + rs.error_code)
print('query_all_stock respond error_msg:' + rs.error_msg)

# 打印结果集
code_list = []
while (rs.error_code == '0') & rs.next():
    # 获取一条记录, 将记录合并在一起
    code_list.append(rs.get_row_data()[0])
print(code_list)

df = pd.DataFrame()
# 获取沪深A股历史K线数据
for code in code_list:
    # 详细指标参数, 参见“历史行情指标参数”章节
    rs = bs.query_history_k_data(code,
                                "date,code,pcfNcfTTM",
                                start_date='2018-06-28',
                                end_date='2018-06-28',
                                frequency="d", adjustflag="3")

    if rs.error_code == '0':
        result = rs.get_data()
        n = result.shape[0]
        if n <= 0:
            continue
        # 删除pe,pcf,pb为0的证券或指数
        if float(result.iloc[0, 2]) != 0:
            if df.empty:
                df = rs.get_data()
            else:
                df = pd.concat([df, rs.get_data()])
        # print(df)

# 结果集输出到csv文件
df.to_csv("D:\\history_A_stock_k_data.csv", index=False)
```

```
print(df)

df['pcfNcfTTM'] = df['pcfNcfTTM'].astype(float)

# 以pcfNcfTTM进行升序排序
df_sortby_pcfNcfTTM = df.sort_values(by='pcfNcfTTM')
df_sortby_pcfNcfTTM.to_csv("D:\\history_A_stock_k_data2.csv",
index=False)
print("A股市场当日市现率最低的证券: " + df_sortby_pcfNcfTTM.iloc[0][1])

# 登出系统
bs.logout()
```

在这里，我很希望刚刚接触量化的小伙伴们多多自己动手写一些策略。目前，市场上免费的 python 接口的证券数据主要是 tushare 和 baostock。之前我也用 tushare，后来发现它多少有些不稳定，现在改用 baostock。从简单的获取数据开始自己的量化之路。