

## 基础不简单之市净率

证券里很多的基础概念都非常有用，如市净率。这些基础概念在不同模型中作为常见的量化因子。了解这些基础概念，最根本的是了解其求解公式（博主是数学系毕业的，惯性思维，请见谅勿喷）。

市净率（Price to book ratio 即 PB），指的是每股股价与每股净资产的比率，即 市净率 = 每股股价 / 每股净资产。

股价代表的是当前市场对该公司的估值，净资产代表该公司实际上的资产净值。所以，市净率能反映出当前估值和实际净值之间的比值。当  $pb = 1$ ，市场对该公司的估值等于其所有资产净值；当  $pb > 1$ ，说明市场给了该只股票大于其本身现值的溢价；当  $pb < 1$ ，说明当前估值小于公司的现值，市场给出了折价的判断。公司被高估，反映市场对该公司有比较高的预期，认为其具有更多潜在价值，更高的成长性，净资产将会出现上涨。但是在中国证券市场经常出现炒“热点”的场景，而且长久不衰。也就是，处于某一个“热点”行业的证券股价会被炒的很高，市净率自然会很高，当某“热点”退热后，股价就会跌回原本的真实估值。同时，公司被低估，市净率小于 1，一方面代表投资的好时机，巴菲特等股神不就是喜欢购买往往被低估的公司，另一方面代表市场对该公司的前景没有信心，认为公司的经营情况将会进一步恶化，资产净值会进一步下跌。

是不是感觉头大了？市净率这个指标本身包含了现实价值和未来预期两部分的信息。如何抉择？一般地来说，如果公司处于高速增长期，市净率比较高是正常的，如果公司已经处于成熟期，增速放缓，或者没有高于市场平均值，且公司资产大量为实物资产，市净率就不应该远高于 1。所以，当在评估高风险企业，企业资产大量为实物资产的企业时，市净率是一个非常重要的指标。

在一般的证券软件见到的市净率指标，不是 pb，而是 pb(MQR)或 pbMQR。这个指标是什么呢？不急，我慢慢道来。MQR 全称是 MRQ 是指 most recent quarter，即最近一个季度。那么，pbMQR 自然表示最近一个季度的市净率。因为市净率的计算，需要参考财务报表中公司的净资产，当然是最新的财务报表更有准确。

假如，博主证券资金不多，喜欢捡便宜，想学学“股神”巴菲特，在全市场中筛选被严重低估的证券（不要打脸，只是“假如”）。如何操作呢？具体代码如下：

```
#!/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,pbMRQ",
                                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pbMRQ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['pbMRQ'] = df['pbMRQ'].astype(float)
# 以pbMRQ进行升序排序
df_sortby_pbMRQ = df.sort_values(by='pbMRQ')
# 存入文件
```

```
df_sortby_pbMRQ.to_csv("D:\\history_A_stock_k_data2.csv",
index=False)
print("当天A股市场pb最低的证券: " + df_sortby_pbMRQ.iloc[0][1])

# 登出系统
bs.logout()
```

代码中 baostock 模块，是需要自己安装的。具体参考它的官网：[www.baostock.com](http://www.baostock.com)

当然，你可以修改代码里面的 code\_list 中证券代码。选择你之前初步筛选过的证券代码集合，再用市净率筛选一遍。最简单的量化就是不同的因子进行不同的组合筛选。