广东金融学院实验报告

课程名称：Python程序基础

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验编号  及实验名称 | 实验三 股票数据获取及分析 | | | 系 别 | 互联网 |
| 姓 名 |  | 学 号 |  | 班 级 |  |
| 实验地点 |  | 实验日期 | 2020.06.18 | 实验时数 | 4 |
| 指导老师 | 唐名华 | 同组其他成员 | 无 | 成 绩 |  |
| 一、实验目的及要求  1、理解Python语言函数应用，能够正确编写以及调用函数。  2、掌握Tushare库的使用，掌握一种股票指标。 | | | | | |
| 二、实验环境及相关情况（包含使用软件、实验设备、主要仪器及材料等）  1．使用软件：Python  2．实验设备：装有Python的联网的个人计算机 | | | | | |
| 三、实验内容及步骤（包含简要的实验步骤流程）  1、编写程序利用Tushare库下载股票数据，自选一个股票指标，对所有股票数据进行分析，根据分析结果推荐一支股票。程序和数据都不用截图的方式。 | | | | | |
| 1. 实验结果（包括程序或图表、结论陈述、数据记录及分析等，可附页）   筛选思路：   1. 首先根据每个股的短期涨幅（这里取3号到17号，11天），与沪深300的涨幅作比较，涨幅小于沪深300的筛去。 2. 根据三个指标进行二次筛选。第一，股票的平均量比，筛选出量比高于1.5，温和放量上涨趋势较为明显。对换手率进行筛选，对于区间的换手率均值需要超过5%，即在市场内相对的热门股票。加上一个基本面的分析，pe动态市盈率不能太高，以pe小于35为指标筛选股票。 3. 最后计算股票的BOLL线，根据收盘价到下轨道线的距离/中轴线到下轨道线的距离的比值进行排序，筛选出最终的结果。   <python>  # 通过接口获取数据的代码  import tushare as ts import pandas as pd import numpy as np import matplotlib.pyplot as plt import time   pro = ts.pro\_api()   def get\_daily\_data(stock\_list):  judge = 0   for k, v in stock\_list.items():  df = pro.daily(ts\_code=v, start\_date='20200603', end\_date='20200617')  df.to\_csv("./daily\_data.csv", header=False, index=False, mode='a')  judge += 1  print(judge)  if judge % 200 == 0:  time.sleep(30)   def get\_daily\_basic(stock\_list):  judge = 0  flag = True  mode = 'w'  for k, v in stock\_list.iteritems():  df = pro.daily\_basic(ts\_code=v, start\_date='20200603', end\_date='20200617')  df.to\_csv("./daily\_data\_basic.csv", header=flag, index=False, mode=mode)  judge += 1  print(judge)  if flag:  mode = 'a'  flag = False  if judge % 200 == 0:  time.sleep(50)   def get\_stock\_ma(stock\_list):  judge = 0  flag = True  mode = 'w'  for k, v in stock\_list.iteritems():  df = ts.pro\_bar(ts\_code=v, start\_date='20200417', end\_date='20200617', ma=[20])  df.to\_csv("./daily\_data\_ma.csv", header=flag, index=False, mode=mode)  judge += 1  print(judge)  if flag:  mode = 'a'  flag = False  if judge % 200 == 0:  time.sleep(50)   if \_\_name\_\_ == '\_\_main\_\_':  # time.sleep(60)  # 获取股票列表  # stock\_list = pro.stock\_basic(fields='ts\_code')['ts\_code']  # get\_daily\_data(stock\_list)  # stock\_list = pd.read\_csv("first\_filter\_stock.csv")['ts\_code']  # get\_daily\_basic(stock\_list)  stock\_list = pd.read\_csv("second\_filter\_stock.csv")['ts\_code']  get\_stock\_ma(stock\_list)  进行数据分析的代码  #%%  import tushare as ts import pandas as pd import numpy as np import matplotlib.pyplot as plt import time  #%%  # 获取pro接口 pro = ts.pro\_api() # 获取沪深300指数，从6月3日到17日的日线 hs300 = pro.index\_daily(ts\_code='000300.SH', start\_date='20200603', end\_date='20200617') # 求出沪深300的涨幅 hs300\_zf = hs300['pct\_chg'].sum()  #%%  # 读取日线数据文件 stock\_data = pd.read\_csv("daily\_data.csv")  #%%  # 对数据进行分组，分组依据 股票代码 stock\_gb = stock\_data.groupby('ts\_code') # 因为停牌所占的比例不大，所以可以选择忽略 # 获取股票11天内的股票涨幅 stock\_zf = stock\_gb.pct\_chg.sum()  #%%  # 第一次筛选，得到区间内跑赢沪深300的股票 stock\_data\_1 = stock\_zf[stock\_zf >= hs300\_zf] # 将筛选出来的股票写入到csv文件中，进行下一步的数据获取 pd.Series(stock\_data\_1.index).to\_csv("first\_filter\_stock.csv", index=False)  #%%  # 读取数据文件，准备第二次的筛选 stock\_data\_1 = pd.read\_csv('daily\_data\_basic.csv') stock\_data\_gb1 = stock\_data\_1.groupby('ts\_code')  #%%  # 求股票的量比均值，区间内量比高于1.5，说明股票温和放量，有一定的上升趋势 stock\_volume\_ratio = stock\_data\_gb1['volume\_ratio'].mean() stock\_volume\_ratio = stock\_volume\_ratio[stock\_volume\_ratio > 1.5]  #%%  # 求股票的pe均值，设定区间大于5小于35 stock\_pe = stock\_data\_gb1['pe\_ttm'].mean() stock\_pe = stock\_pe[(stock\_pe <= 30) & (stock\_pe >= 5)]  #%%  # 求股票的换手率，找出区间内资本热衷的股票 stock\_turnover = stock\_data\_gb1['turnover\_rate\_f'].mean() stock\_turnover = stock\_turnover[stock\_turnover > 5]  #%%  # 将第二次筛选出来的股票写入到文件中，获取第三次所需的数据 pd.Series(list(set(stock\_pe.index).intersection(set(stock\_volume\_ratio.index)).intersection(stock\_turnover.index))).to\_csv("second\_filter\_stock.csv", header=['ts\_code'],index=False)  #%%  # 读取数据文件 stock\_data\_2 = pd.read\_csv('daily\_data\_ma.csv') # 删除一些多余的数据列 stock\_data\_2.drop(['open', 'high', 'low', 'pre\_close', 'pct\_chg', 'vol', 'amount', 'ma\_v\_20'], axis=1, inplace=True) # 去除空值 stock\_data\_2.dropna(how='any', axis=0, inplace=True) # MD=平方根N日的（C－MA）的两次方之和除以N # 计算标准差 # 首先计算每天的收盘价和均线的差的平方 poor = (stock\_data\_2['close']-stock\_data\_2['ma20'])\*\*2 stock\_data\_2['poor'] = poor # 从17号往前推20个交易天，取得他们的交易数据(上面求出的差的平方)，即五月二十号开始 stock\_data\_2 = stock\_data\_2[stock\_data\_2['trade\_date'] >= 20200520] # 计算差的和,再开放最后除以20（天） stock\_data\_gb2 = stock\_data\_2.groupby('ts\_code') md = stock\_data\_gb2['poor'].sum()\*\*0.5/20  #%%  # 取出17号的数据，并将标准差填充进去 stock\_data\_2 = stock\_data\_2[stock\_data\_2['trade\_date'] == 20200617] # 修改dataframe的索引 stock\_data\_2 = stock\_data\_2.set\_index(['ts\_code']) stock\_data\_2['md'] = md  #%%  # 计算上下轨道线 stock\_data\_2['dn']=stock\_data\_2['ma20']-stock\_data\_2['md']\*2 stock\_data\_2['up']=stock\_data\_2['ma20']+stock\_data\_2['md']\*2  #%%  # 求出当天收盘价到下轨道线的距离与中轴线到下轨道线距离的比例 stock\_data\_2['judge'] = (stock\_data\_2['close']-stock\_data\_2['dn'])/(stock\_data\_2['ma20']-stock\_data\_2['dn'])  #%%  stock\_data\_2  #%%  ans = stock\_data\_2.sort\_values(by=['judge'])  #%%  ans.head()  </python> | | | | | |
| 五、实验总结（包括心得体会、问题回答及实验改进意见，可附页） | | | | | |
| 六、教师评语  1、完成所有规定的实验内容，实验步骤正确，结果正确；  2、完成绝大部分规定的实验内容，实验步骤正确，结果正确；  3、完成大部分规定的实验内容，实验步骤正确，结果正确；  4、基本完成规定的实验内容，实验步骤基本正确，所完成的结果基本正确；  5、未能很好地完成规定的实验内容或实验步骤不正确或结果不正确。  评定等级：  签名：唐名华 | | | | | |