

财经数据接口包 tushare 的使用（一）

Tushare 是一款开源免费的金融数据接口包，可以用于获取股票的历史数据、年度季度报表数据、实时分笔数据、历史分笔数据，本文对 tushare 的用法，已经存在的一些问题做一些介绍。

一：安装 tushare

为避免由于依赖包缺失导致安装失败，请先安装 anaconda，百度网盘地址：

链接：<http://pan.baidu.com/s/1qYDQUGs> 密码：6wq8

安装直接一直下一步即可

安装完成之后，anaconda 会自动配置环境变量，直接就可以用了，cmd 打开命令行窗口，使用命令

```
pip install tushare
```

即可安装 tushare

二：tushare 的使用

1、获取历史数据之 get_hist_data

参数说明：

- **code**: 股票代码，即 6 位数字代码，或者指数代码（sh=上证指数 sz=深圳成指 hs300=沪深 300 指数 sz50=上证 50 zxb=中小板 cyb=创业板）
- **start**: 开始日期，格式 YYYY-MM-DD
- **end**: 结束日期，格式 YYYY-MM-DD
- **ktype**: 数据类型，D=日 k 线 W=周 M=月 5=5 分钟 15=15 分钟 30=30 分钟 60=60 分钟，默认为 D
- **retry_count**: 当网络异常后重试次数，默认为 3

-
- **pause**: 重试时停顿秒数, 默认为 0

返回值说明:

- **date**: 日期
- **open**: 开盘价
- **high**: 最高价
- **close**: 收盘价
- **low**: 最低价
- **volume**: 成交量
- **price_change**: 价格变动
- **p_change**: 涨跌幅
- **ma5**: 5 日均价
- **ma10**: 10 日均价
- **ma20**: 20 日均价
- **v_ma5**: 5 日均量
- **v_ma10**: 10 日均量
- **v_ma20**: 20 日均量
- **turnover**: 换手率[注: 指数无此项]

调用方法:

例一:

```
import tushare as ts
data=ts.get_hist_data('300032')
print(data)
```

结果展示:

	open	high	close	low	volume	price_change	p_change \
date							
2017-11-22	13.89	13.96	13.95	13.77	19670.56	-0.02	-0.14
2017-11-21	13.97	13.98	13.97	13.77	23196.87	-0.01	-0.07
2017-11-20	13.90	14.02	13.98	13.68	23114.10	0.01	0.07

2017-11-17	13.87	13.97	13.97	13.67	39828.91	0.00	0.00
2017-11-16	13.98	14.09	13.97	13.86	18029.02	-0.04	-0.29
2017-11-15	14.37	14.37	14.01	13.88	31731.32	-0.37	-2.57
2017-11-14	14.00	14.43	14.38	14.00	49168.37	0.32	2.28

	ma5	ma10	ma20	v_ma5	v_ma10	v_ma20	turnover
date							
2017-11-22	13.968	14.046	14.044	24767.89	26789.33	28602.29	0.26
2017-11-21	13.980	14.048	14.054	27180.04	27617.60	28540.55	0.31
2017-11-20	14.062	14.043	14.064	32374.34	28650.48	28455.20	0.31
2017-11-17	14.078	14.038	14.083	31644.05	29352.32	28141.07	0.53
2017-11-16	14.114	14.039	14.099	28582.57	29823.22	27169.73	0.24
2017-11-15	14.124	14.042	14.118	28810.76	30467.91	27007.47	0.42
2017-11-14	14.116	14.046	14.132	28055.16	30117.38	26536.15	0.66

请读者注意，文档中的注释并不准确，使用该接口并不能获取股票自上市以来的所有日线数据，例如这里举的例子，金龙机电这只股票(2009年12月25日上市)。

请再看例子：

例二：

```
import tushare as ts
data=ts.get_hist_data('300032', start='2011-01-01', end='2011-05-01')
print(data)
```

返回结果是：

```
Empty DataFrame
Columns: [open, high, close, low, volume, price_change, p_change, ma5, ma10, ma20, v_ma5, v_ma10, v_ma20, turnover]
Index: []
```

可以看到，这几行代码返回的数据为一个空的 dataframe，这是网页文章中没有详细说明的问题在这里指出。如何解决这个问题，请关注微信公众号【数据之佳】，注意是才子佳人的佳，该公众号分享量化分析模型，讲解一些比较好的模型，工具等等，tushare 的使用公众号中有更详细的介绍。公众号也会讲解如何用其他方式获取更稳定可靠的数据

2、获取历史数据之 get_h_data

作者提供另外一个函数 `get_h_data` 用于获取股票历史数据，该函数的调用网页文档没有给出例子，在这里依然用上面的例子来作测试：

例三：

```
import tushare as ts
data=ts.get_h_data('300032')
print(data)
```

返回结果：

	open	high	close	low	volume	amount
date						
2017-11-22	13.89	13.96	13.95	13.77	1967056.0	27282987.0
2017-11-21	13.97	13.98	13.97	13.77	2319687.0	32194061.0
2017-11-20	13.90	14.02	13.98	13.68	2311410.0	32033838.0
2017-11-17	13.87	13.97	13.97	13.67	3982891.0	54911258.0
.....						
2016-11-24	17.31	17.42	17.17	17.15	3799600.0	66045345.0
2016-11-23	17.37	17.51	17.27	17.24	5008208.0	87687070.0
2016-11-22	17.37	17.41	17.40	17.20	4536796.0	79050365.0

[246 rows x 6 columns]

在不指定开始时间和结束时间时，该函数默认返回最近一年的日线数据，返回的数据与 `get_hist_data` 不同的是，该函数只返回开盘价（open）、最高价（high）、收盘价（close）、最低价（low）、成交量（volume）、成交金额（amount）六列

同样的，测试一下该接口能不能获取更早一些的数据：

```
import tushare as ts

data=ts.get_h_data('300032', start='2011-01-01', end='2011-05-01')

print(data)
```

返回结果:

```
[Getting data:]#           open  high  close   low    volume    amount
date
2011-04-29  14.18  14.41  14.41  13.96   444287.0  10930304.0
2011-04-28  14.60  14.66  14.10  13.86  2550197.0  63069508.0
2011-04-27  14.50  14.67  14.44  14.26   457503.0  11427143.0
2011-04-26  14.82  14.82  14.41  14.31   628685.0  15732398.0
2011-01-04  13.14  13.23  13.21  12.84  1099825.0  24814884.0
[77 rows x 6 columns]
```

可以看到, 相比于 `get_hist_data`, 该函数能够返回较早一些的数据, 但没有换手率, 均线数据这些指标, 当然, 我们可以获取到数据之后, 自己写个脚本计算均线这些信息, 但是如果要自己计算换手率, 就必须知道股票的市值是多少, 这两个函数都没有返回关于股票市值的信息, 因此如果要自己计算换手率, 就得从其他地方获取股票每个交易日的市值信息。

3、获取历史数据之: `get_k_data`

作者提供了第二个获取K线数据的函数, `get_k_data`, 参数与 `get_hist_data` 相同。

例四:

```
import tushare as ts

data=ts.get_k_data('300032')

print(data)
```

返回结果:

```
   date    open  close   high   low  volume    code
0  2015-02-02  12.914  13.742  14.086  12.914  56295.0  300032
1  2015-02-03  13.742  14.377  14.500  13.742  63588.0  300032
```

2	2015-02-04	14.412	14.510	14.761	13.791	35558.0	300032
3	2015-02-05	14.520	14.471	14.751	14.224	34088.0	300032
4	2015-02-06	14.648	14.343	14.648	13.845	35039.0	300032
5	2015-02-09	14.372	14.062	14.549	13.904	24195.0	300032

与前两个函数相比，这个函数获取数据的速度很明显要快很多，而且可以返回每一只股票从上市开始到当前交易日的所有日线数据，这个有点是前两个函数都不具备的，读者可以自己验证一下，`get_h_data` 可以返回比 `get_hist_data` 更早一些的数据，但是对于有些股票依然不能获取很早的数据，更重要的是，如果批量 3000 多只股票的数据，前两个都不如 `get_k_data` 稳定，我一开始搜集数据的时候，为了获取更全面的数据，用的是 `get_hist_data`，但是批量获取，反反复复跑脚本，没有一次是能够将数据完整获取完的，每一次都是中途就挂了，读者可以自己测试一下，间隔时间长一些应该没问题，不过我后来已经知道其他的解决办法，既能够获取全面的数据，同时也不必担心速度的问题，怎么获取一样会在微信公众号【数据之佳】介绍，这里先介绍 `tushare`

4、获取实时行情数据 `get_today_all()`

返回值说明：

- **code**: 代码
- **name**: 名称
- **changepercent**: 涨跌幅
- **trade**: 现价
- **open**: 开盘价
- **high**: 最高价
- **low**: 最低价
- **settlement**: 昨日收盘价
- **volume**: 成交量
- **turnoverratio**: 换手率
- **amount**: 成交额
- **per**: 市盈率

- **pb**: 市净率
- **mktcap**: 总市值
- **nmc**: 流通市值

该函数没有参数，直接调用即可

测试：

例五：

```
import tushare as ts
data=ts.get_today_all()
print(data)
```

返回值：

```
[Getting data:]#####
#####      code  name  changepercent  trade  open  high  low
settlement \
0    603999  读者传媒      -0.622   7.99  8.03  8.11  7.92      8.04
1    603998  方盛制药      -2.535  12.69 13.00 13.13 12.58      13.02
2    603997  继峰股份      -1.529  11.59 11.77 11.86 11.50      11.77
.....
      volume  turnover ratio  amount  per  pb  mktcap \
0    2208414.0      0.95851 17697355.0 27.552 2.759 4.602240e+05
1    2733900.0      1.01565 34976989.0 79.313 5.471 5.468584e+05
2    1848500.0      1.08099 21572699.0 19.644 4.428 7.301700e+05
      nmc
0    1.840896e+05
1    3.415859e+05
2    1.981890e+05
.....
[3446 rows x 15 columns]
```

限于篇幅，这里只贴出前三行数据，`get_today_all()` 获取了所有股票的当前行情数据，但是获取一次数据的耗时比较长，读者可以自己在实时行情和盘后自己测试一下，看看会不会挂，整体延时有多久等等

5、历史分笔数据之：get_tick_data

参数说明：

code：股票代码，即 6 位数字代码

date：日期，格式 YYYY-MM-DD

retry_count : int, 默认 3, 如遇网络等问题重复执行的次数

pause : int, 默认 0, 重复请求数据过程中暂停的秒数，防止请求间隔时间太短出现的问题

该函数返回指定日期的历史分笔数据，但由于历史分笔数据很耗空间，所以一般服务器都不会存储很长时间的历史分笔数据，该接口也一样，并不能获取从上市以来的所有日期的分笔数据。

例六：

```
import tushare as ts  
data=ts.get_tick_data('300032', '2017-11-01')  
print(data)
```

返回结果：

	time	price	change	volume	amount	type
0	15:00:03	14.05	-0.01	371	521255	卖盘
1	14:57:03	14.06	0.01	1	1406	买盘
2	14:56:57	14.05	--	92	129260	卖盘
3	14:56:54	14.05	--	20	28100	卖盘
.....						
1526	09:30:36	14.10	-0.02	159	224190	中性盘
1527	09:30:06	14.12	14.12	7	9884	买盘

[1528 rows x 6 columns]

6、实时分笔数据之 get_realtime_quotes

参数说明：

- **symbols**: 6 位数字股票代码，或者指数代码（sh=上证指数 sz=深圳成指
hs300=沪深 300 指数 sz50=上证 50 zxb=中小板 cyb=创业板） 可输入的类型：**str**、**list**、**set** 或者 **pandas** 的 **Series** 对象

例八：

```
import tushare as ts
```

```
data=ts.get_realtime_quotes('300032')
```

```
print(data)
```

```
   name  open pre_close  price   high   low   bid   ask  volume \
0  金龙机电  13.890    13.970  13.950  13.960  13.770  13.940  13.950  1967056
      amount  ...      a2_p a3_v   a3_p a4_v   a4_p a5_v   a5_p  \
0  27282987.240  ...    13.960  741   13.970  547   13.980  332   13.990
      date      time   code
0  2017-11-22  16:28:03  300032
[1 rows x 33 columns]
```

该接口返回的数据量较小，还是比较快的，读者可以自己在实时行情中和盘后自己测试一下

6、当日历史分笔之 get_today_ticks

参数说明：

code: 股票代码，即 6 位数字代码

retry_count : int, 默认 3, 如遇网络等问题重复执行的次数

pause : int, 默认 0, 重复请求数据过程中暂停的秒数，防止请求间隔时间太短出现的问题

例九

```
import tushare as ts
```

```
data=ts.get_today_ticks('300032')
```

```
print(data)
```

返回值

```
[Getting data:]#####      time  price pchange  change  volume  amount type
0   15:00:03  14.36  +2.79   0.00   938  1346968  卖盘
1   14:57:00  14.36  +2.79   0.01     2    2872  卖盘
2   14:56:57  14.35  +2.72  -0.01    27   39463  卖盘
.....
984 09:25:03  13.89  -0.57   0.00    27   37503   0
[985 rows x 7 columns]
```

该接口返回当前日期，当前时刻的所有粉笔成交数据，相比于上一个接口，速度慢很多，也请读者自己测试

7、大单交易数据之 get_sina_dd

获取大单交易数据，默认为大于等于 400 手，数据来源于新浪财经。

参数说明：



- **code**: 股票代码，即 6 位数字代码
- **date**: 日期，格式 YYYY-MM-DD
- **vol**: 手数，默认为 400 手，输入数值型参数
- **retry_count** : int, 默认 3, 如遇网络等问题重复执行的次数
- **pause** : int, 默认 0, 重复请求数据过程中暂停的秒数，防止请求间隔时间太短出现的问题

返回值说明：

- **code**: 代码
- **name**: 名称
- **time**: 时间
- **price**: 当前价格
- **volume**: 成交手

-
- **preprice** : 上一笔价格
 - **type**: 买卖类型【买盘、卖盘、中性盘】

例 10:

```
import tushare as ts  
data=ts.get_sina_dd('300032','2017-11-1')  
print(data)
```

返回值

	code	name	time	price	volume	preprice	type
0	300032	金龙机电	14:45:54	13.97	91900	14.00	卖盘
1	300032	金龙机电	10:51:36	14.10	57488	14.11	卖盘
2	300032	金龙机电	10:19:42	14.05	50000	14.06	卖盘
3	300032	金龙机电	09:38:36	14.03	42800	14.02	买盘

借助于 tushare，其实我们已经可以做一些量化分析和策略了，但是 tushare 在实际使用中，有一些不经如人意的地方，我在使用中遇到的问题，同时也自己优化了一一些地方，实在不满意的数据，从其他地方用自己想办法获取，对于 tushare 已经可以做得很好的就不洗自己再去做了，想要了解怎样获取更全面的数据，学习如何将机器学习算法和数据分析方法运用到自己的量化模型当中，请关注微信公众号【**数据之佳**】，我们将不定期更新文章



殺器之海