

# RQData安装教程

## RQData简介

RQData SDK是一个面向机构的商用版Python金融数据工具包。通过极简的本地安装方式，用户即可享用RQData强大的数据查询访问功能。它集成了简单高效的API接口，用户可快速调用丰富整齐的量化金融数据，最大限度地免除了数据搜索、清洗的烦恼，加速投研及投资的决策周期。

Ricequant为您提供数据包括：中国A股、ETF、中国期货（股指、国债、商品期货）的所有基本信息和每日市场数据/分钟数据，A股/期货的tick数据，以及A股上市的财务数据和场内基金数据。

\*如一个月试用期结束，请添加微信号RQData延长试用时长。

## RQData的一些亮点

- 本地化API调用，可与用户自由环境/数据组合使用，灵活度非常高
- 数据调取速度极快，**不限制流量**(注：正式版)
- 自动化数据清洗，更新及时，更早地获取到最新的数据
- API设计专业简洁，符合量化投研习惯，上手容易，提升查询数据效率
- 支持Python、Matlab和Excel API调用，与机构自身投研环境融为一体

## RQData的安装与使用

### 第一步 安装Ricequant的RQData

我们会为您提供使用RQData的License Key以及安装文件。如果您尚未获得，还请与我们的商务或技术支持联系。安装RQData，您的电脑上需要已配置的**python**环境，python2或者python3（任意一个都可以）。

#### 1.Windows 系统

##### 步骤一：安装RQData

```
pip install --extra-index-url https://rquser:ricequant99@py.ricequant.com/simple/ rqdatac==1.0.0a22
```



##### 步骤二：配置License

我们已将您的License Key配置在安装脚本中，您无需另外输入。您只需双击运行米筐提供的安装脚本附件中的make.bat脚本文件即可。（注意：1 杀毒软件提示时请选择允许运行；2 由于make.bat生成环境变量，已打开的终端需要重新打开才能获得新的环境变量。）

#### 2.Mac 系统

##### 步骤一：安装RQData 打开终端并输入如下code(rqdatac前面要空格)

```
pip install --extra-index-url https://rquser:ricequant99@py.ricequant.com/simple/ rqdatac==1.0.0a22
```

##### 步骤二：配置License

打开终端，找到系统所对应的文件夹。将米筐提供的附件中的make.sh复制到系统文件夹中。在终端中输入

```
source make.sh
```

即可完成Mac上的RQData配置

#### 3.安装相关的视频示范

- windows系统安装RQData
- Mac系统安装RQData
- anaconda安装RQData
- RQData使用范例

## 第二步 开始使用RQData提取数据

打开python或者spyder和pycharm等IDE

### 1.初始化RQData

键入如下代码进行初始化

```
import rqdatac as rq
from rqdatac import *
rq.init()
```

### 2.使用RQData调用数据

所有的函数列表及使用说明可以参考

- [RQData的API文档](https://www.ricequant.com/doc/rqdata-institutional) (https://www.ricequant.com/doc/rqdata-institutional)
- [财务数据文档](https://www.ricequant.com/fundamentals) (https://www.ricequant.com/fundamentals)

RQData使用十分方便，API中的数据调用函数一律采用如下形式

```
rq.function(para1,para1,para3...)
```

样例：

#### 1.调取历史数据

我们要调取股票000001的2018-3-23到2018-3-12日的tick行情数据

```
rq.get_price('000001.XSHE','2018-3-23','2018-3-23','tick')
```

```
In [6]: rq.get_price('000001.XSHE','2018-3-23','2018-3-23','tick')
Out[6]:
```

	trading_date	open	last	high	low	prev_close
datetime						
2018-03-23 09:15:00	2018-03-23	11.25	11.6600	0.00	0.00	11.66
2018-03-23 09:15:09	2018-03-23	11.25	11.6600	0.00	0.00	11.66
2018-03-23 09:15:18	2018-03-23	11.25	11.6600	0.00	0.00	11.66
2018-03-23 09:15:27	2018-03-23	11.25	11.6600	0.00	0.00	11.66
2018-03-23 09:15:36	2018-03-23	11.25	11.6600	0.00	0.00	11.66
2018-03-23 09:15:45	2018-03-23	11.25	11.6600	0.00	0.00	11.66
2018-03-23 09:15:54	2018-03-23	11.25	11.6600	0.00	0.00	11.66
2018-03-23 09:16:03	2018-03-23	11.25	11.6600	0.00	0.00	11.66
2018-03-23 09:16:12	2018-03-23	11.25	11.6600	0.00	0.00	11.66
2018-03-23 09:16:21	2018-03-23	11.25	11.6600	0.00	0.00	11.66
2018-03-23 09:16:30	2018-03-23	11.25	11.6600	0.00	0.00	11.66
2018-03-23 09:16:39	2018-03-23	11.25	11.6600	0.00	0.00	11.66
2018-03-23 09:16:48	2018-03-23	11.25	11.6600	0.00	0.00	11.66

#### 2.调取财务数据

获取单只股票过去两个报告期的净利润

```
q = query(financials.income_statement.net_profit, financials.announce_date).filter(financials.stockcode.in_(['000002.XSHE']))
get_financials(q, '2016q3', '2q')
```

```
[In]
q = query(financials.income_statement.net_profit, financials.announce_date).filter(financials.stockcode.in_(['000002.XSHE']))
[In]
get_financials(q, '2016q3', '2q')
[Out]
      net_profit announce_date
2016q3  1.12903e+10      20161028
2016q2  7.09463e+09      20160822
```

### 3. 获取当前的实时行情数据

```
current_snapshot('000001.XSHE')
```

```
In [11]: current_snapshot('000001.XSHE')
Out[11]: Tick(ask_vols: [58507, 46300, 168580, 130300, 97700], asks: [10.91, 10.92, 10.93, 10.94, 10.95], bid_vols: [87167, 648419, 1029400, 305900, 222800], bids: [10.9, 10.89, 10.88, 10.87, 10.86], datetime: 2018-03-30 16:29:03, high: 11.05, last: 10.9, low: 10.88, open: 11.04, open_interest: None, order_book_id: 000001.XSHE, prev_close: 11.05, prev_settlement: None, total_turnover: 823438180, volume: 75217368)
```

## 3. 数据导出

如果想要将数据导出到本地，只需要通过以下代码即可实现：

```
import pandas as pd
df = pd.DataFrame(xxxx)
df.to_csv('xxx.csv')
```

其中，xxx即为想要导出的数据。

```
In [12]: data=rq.get_price('平安银行','2018-3-23','2018-3-23')
...: import pandas as pd
...: df = pd.DataFrame(data)
...: df.to_csv('payh_20180323.csv')
```

## 附录 应用举例

利用股价序列研究同行业股票之间的协整关系

```
import rqdatac
from rqdatac import *
import pandas as pd
import numpy as np
import statsmodels.api as sm
import seaborn as sns

rq.init()

## 找到能源行业所有股票列表
selected_plate=industry('J66')
stock_list = list(selected_plate)
# 调取该行业近一年股票收盘价数据
dataframe = get_price(stock_list, start_date="2017-01-01", end_date="2017-12-31", adjust_type='internal')['ClosingPx']
n = dataframe.shape[1]
# 初始化p值矩阵
pvalue_matrix = np.ones((n, n))
# 抽取列的名称
keys = dataframe.keys()
# 初始化强协整组
pairs = []
for i in range(n):
    # 对于大于i的j
    for j in range(i+1, n):
        # 获取相应的两只股票的价格Series
```

```

stock1 = dataframe[keys[i]]
stock2 = dataframe[keys[j]]
# 分析它们的协整关系
result = sm.tsa.stattools.coint(stock1, stock2)
# 取出并记录p值
pvalue = result[1]
pvalue_matrix[i, j] = pvalue
if pvalue < 0.05:
    # 记录股票对和相应的p值
    pairs.append((keys[i], keys[j], pvalue))

```

#画出p值热力图

```
sns.heatmap(1-pvalue_matrix, xticklabels=stock_list, yticklabels=stock_list, cmap='RdYlGn_r', mask = (pvalue_matrix == 1))
```

#打印出强协整关系的股票对

```
print (pairs)
```

```

[('600926.XSHG', '000001.XSHE', 0.023924807277559492), ('601328.XSHG', '601166.XSHG', 0.04385563239441625),
('600908.XSHG', '002142.XSHE', 0.03535407757208592), ('600908.XSHG', '000001.XSHE', 0.041866075108510065), ('603323.XSHG',
'002142.XSHE', 0.008876580057220657), ('603323.XSHG', '000001.XSHE', 0.023333234646489256), ('601997.XSHG', '601939.XSHG',
0.03940235138850956), ('601997.XSHG', '600036.XSHG', 0.04865320456006078), ('601997.XSHG', '600919.XSHG',
0.027763228265731313), ('002142.XSHE', '601128.XSHG', 0.04629509505816596), ('002807.XSHE', '601398.XSHG',
0.03342130798008133)]

```

