

RiceQuant 米筐科技

◀ 量化投研RQSDK解决方案 ▶

让 投 资 更 高 效 ， 让 风 险 更 透 明

业务总览

RQSDK 集合了金融数据 API-RQData、多品种回测框架 RQAlpha、因子投研工具 RQFactor、股票组合优化器 RQOptimizer。是一套5分钟开箱即用、专注于解决量化投研的工具。



- **RQData**: 懂金融的数据API，提供数据质量高、API易用、开箱即用无需部署的最佳数据API方案。
- **RQFactor**: 提供快捷因子编写工具，并提供全面的因子分析工具。

- **RQOptimizer**: 支持丰富的目标函数、约束条件与便捷的选股功能，实现股票组合优化。
- **RQAlpha-Plus**: 提供基于事件驱动、贴近实战的多品种的回测引擎。

目录

CONTENT

R I C E Q U A N T



第一部分

RQData

金融数据API



第二部分

RQFactor

因子投研



第三部分

RQOptimizer

股票组合优化器



第四部分

RQAlpha-Plus

回测引擎



第五部分

RQSDK

优势

— Part One ▶▶▶

RQData 金融数据 API

数据、接口种类

丰富的数据种类

A股数据：高频快照、历史行情、行业分类、指数构成、公司行为、技术指标因子等

债券数据：基础数据、估值数据、其他衍生指标等

期权数据：ETF/股指/商品期权全品种支持，上市以来历史数据、高频快照

公募基金数据：基金净值、基金持仓、基金评级、基金衍生数据等

期货数据：金融/商品期货全品种支持，历史数据、高频快照、主力连续合约

可转债数据：高频快照、历史行情、现金流数据、转股信息、含权信息等

风险因子数据：因子暴露度、因子收益率、因子协方差矩阵、因子特异风险等

宏观经济数据：CPI、PPI、货币供应、准备金率等

财务数据：三大表、衍生财务指标，提供PIT数据结构

舆情、电商数据（数据合作方提供）：天猫、京东销售额，雪球、东财舆情数据

多种数据接口



Python



Matlab



RESTful

围绕投研交易设计的API

API围绕投研交易设计，一行代码快速调用。

01

获取某一股票当前快照数据

```
[In] current_snapshot('000001.XSHE')
```

```
[Out]
```

```
Tick(ask_vols: [25400, 15500, 12300, 39985, 16200], asks: [13.7, 13.71, 13.72, 13.73, 13.74], bid_vols: [1050, 9300, 172301, 691800,
```

02

获取某一期货当前快照数据

```
In [22]: current_snapshot('RB2010')
```

```
Out[22]: Tick(ask_vols: [158, 655, 954, 247, 373], asks: [3775.0, 3776.0, 3777.0, 3778.0, 3779.0], bid_vols: [25, 513, 90, 56, 2214]
```

03

获取多个当前快照数据

```
In [22]: current_snapshot('RB2010')
```

```
Out[22]: Tick(ask_vols: [158, 655, 954, 247, 373], asks: [3775.0, 3776.0, 3777.0, 3778.0, 3779.0], bid_vols: [25, 513, 90, 56, 2214]
```

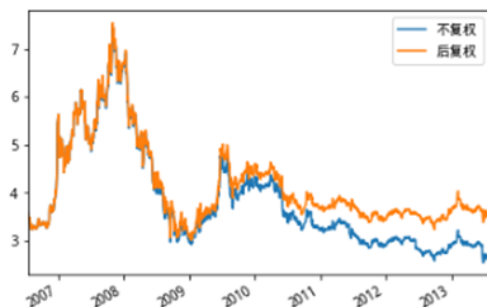
04

更多范例

数据清洗

```
In [10]: import numpy as np
import pandas as pd
sd = '20060716'
ed = '20130730'
stock = '601988.XSHG'
a = get_price(stock, start_date=sd, end_date=ed, frequency='1d', adjust_type='none')['close']
b = get_price(stock, start_date=sd, end_date=ed, frequency='1d', adjust_type='post')['close']
df = pd.DataFrame({'不复权':a, '后复权':b})
df.plot()
```

Out[10]: <matplotlib.axes._subplots.AxesSubplot at 0x7f3ec7925908>



提供贴近实操的数据清洗，
大幅缩短用户处理数据的时间。如：

- 股票前后复权处理
- 财务数据Point-in-Time处理，避免使用未来数据
- 特殊期货合约处理：主力连续合约、指数连续合约

```
[In]
get_pit_financials(['operating_revenue', 'total_assets'], quarter='2018q2', order_book_ids='000048.XSHE')
[Out]
```

order_book_id	end_date	info_date	if_adjusted	accounting_standards	is_complete	enterprise_type	total_assets	operating_revenue
000048.XSHE	2018-06-30	2018-08-31	0	1 1	general_enterprise	4112194743.4400	1063669669.0400	
	2018-06-30	2018-10-31	0	1 1	general_enterprise	4209492787.7000	1060486863.8900	

米筐风险因子数据

提供高质量的米筐风险因子数据。

提供每日更新的因子数据

风格因子暴露度

个股对指数贝塔

因子收益率

因子协方差

特异收益率

特异风险

调用因子数据并计算累计收益率

定义所需的风格因子类型

```
style_factors = ['beta', 'book_to_price', 'earnings_yield', 'growth', 'leverage',  
'liquidity', 'momentum', 'non_linear_size', 'residual_volatility', 'size' ]
```

取得因子收益率

```
factor_return = rqdatac.get_factor_return( 20100101 , 20191204 , style_factors)
```

计算累计收益率

```
cum_factor_return = ( 1 + factor_return).cumprod() - 1
```


— Part Two ▶▶▶

RQFactor因子投研

因子数据

• • 支持丰富的米筐因子库 • •

因子数据	因子数量	备注
财务数据因子	160	利润表、资产负债表、现金流量表等
衍生财务指标因子	300+	估值相关、经营衍生、现金流衍生、财务衍生、成长衍生以及MRQ、TTM、LYR等指标
宏观经济因子	2563	PPI、CPI、PMI、GDP统计，社融规模，公共预算，政府债务等经济指标
Alpha101	101	WorldQuant 的101个技术指标
风险因子	39	风格因子、市场联动、行业因子。包含因子暴露度、因子收益率、特异收益率等
米筐技术指标	140	能量指标-市场强弱、人气意愿，超买超卖指标-相对强弱、市场能量，均线类指标-指数平滑移动均线等
合计	3300+	

因子编写

简便的因子编写功能：

- ① 时间序列算子 ② 横截面算子 ③ 四则运算

```
(rqsdk) C:\Users\ricequant>python
Python 3.8.3 (default, May 19 2020, 06:50:17) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> from rqfactor import *
>>> from rqfactor.indicators import *
>>> # 以自由现金流与总市值之比衡量公司的估值
>>> free_cash_flow = Factor('fcff_ttm')
>>> market_cap = Factor('market_cap_2')
>>> fcfp = free_cash_flow / market_cap
>>> # 时间序列上进行指数平滑
>>> fcfp = EMA(fcfp, 252)# 行业中性化
>>> fcfp = INDUSTRY_NEUTRALIZE(fcfp)
>>> fcfp
CrossSectional(industry_neutralize, EMA(true_divide(Factor('fcff_ttm'), Factor('market_cap_2'))))
```


因子检验

全面的因子分析结果

- 不限因子的资产类型和数据频率;
- 支持传入自定义收益率;
- 支持传入自定义资产行业;
- 对于多日调仓的场景, 使用滚动调仓计算 IC 和因子收益, 从而减少路径依赖;
- 通过构造管道进行数据处理和分析计算, 方便用户反复地进行检验和分析;
- 可单独计算 IC 分析/分组分析/因子收益率, 自定义输出结果更有针对性;
- 支持输出预处理后的因子值以及分组组别, 处理结果更加透明。

#构建管道

```
engine = FactorAnalysisEngine()
```

#构建预处理项

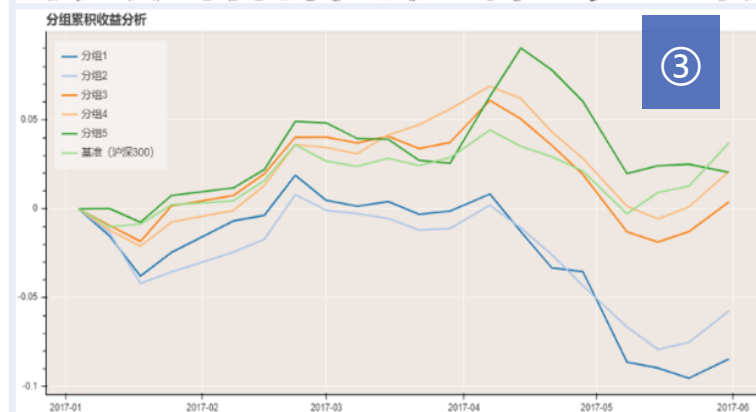
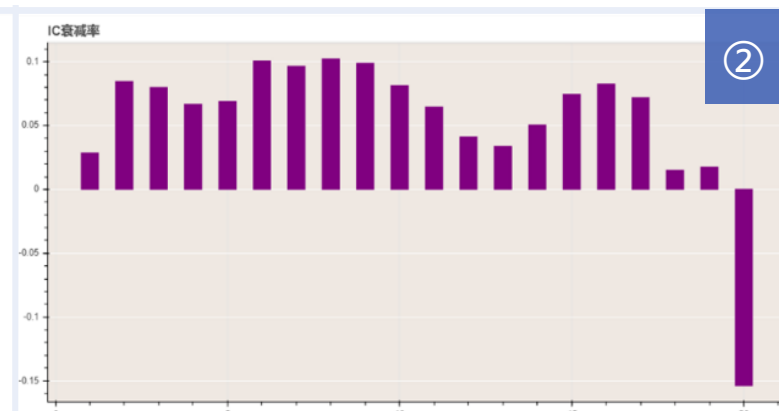
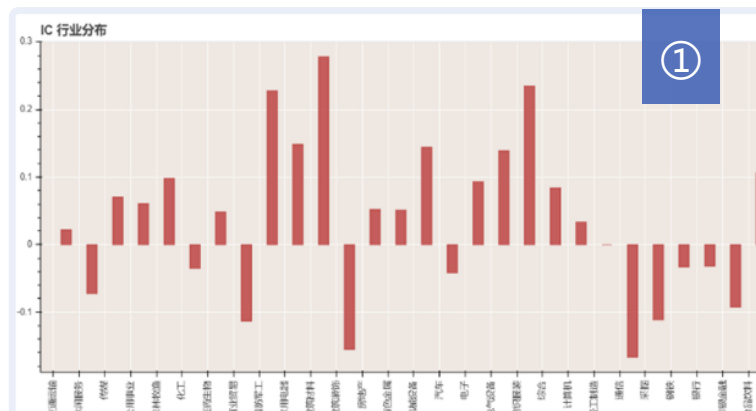
```
engine.append(('winzORIZATION-mad', WinzORIZATION(method='mad')))
```

#构建分析器

```
engine.append(('rank_ic_analysis', ICAnalysis(rank_ic=True, industry_classification='sws')))
```

#执行计算

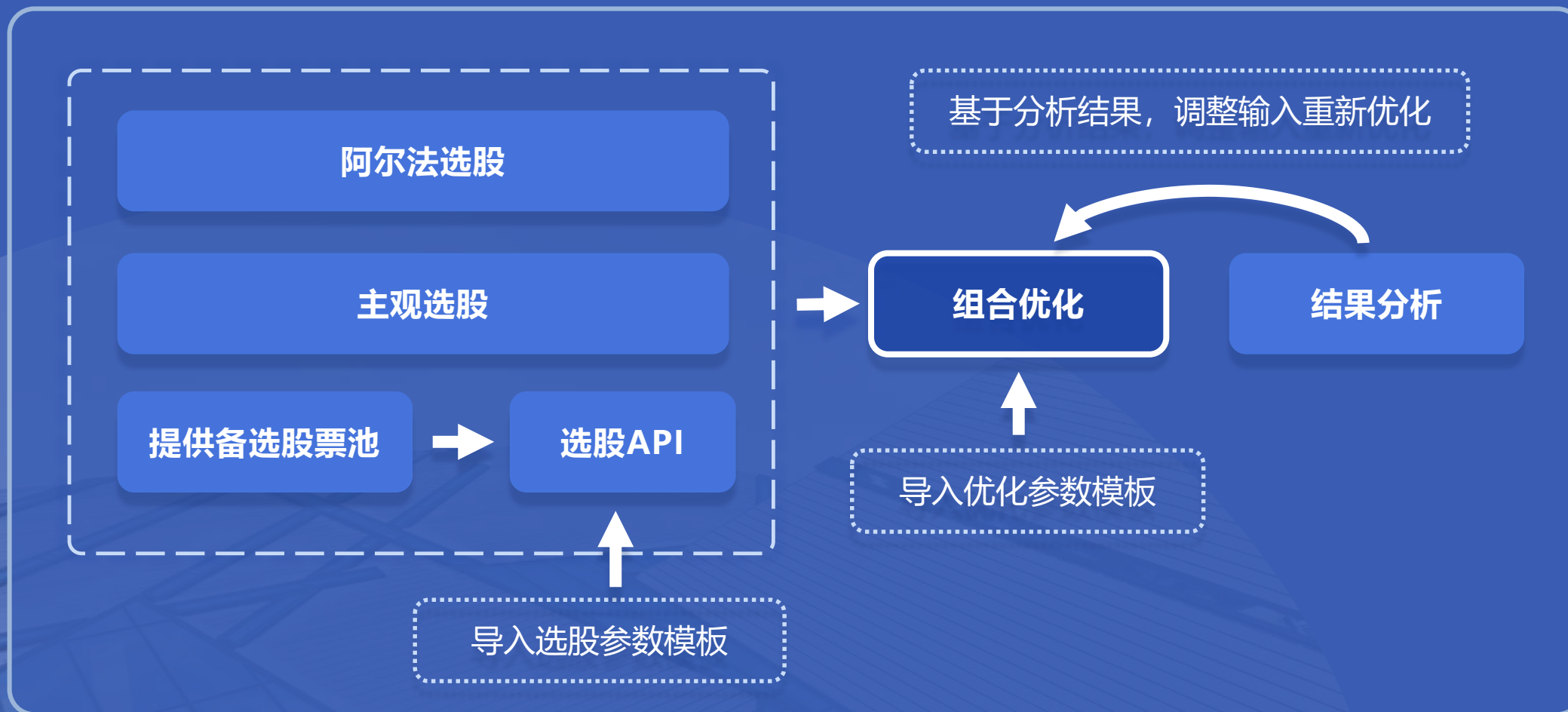
```
result = engine.analysis(df, returns, ascending=True, periods=1, keep_preprocess_result=True)
```



— Part Three ▶▶▶

RQOptimizer股票组合优化器

使用流程



功能

提供丰富的目标函数和约束条件、便捷的选股功能。

目标函数

- 波动率最小化
- 追踪误差最小化
- 均值方差
- 信息率最大化
- 风格偏离最小化
- 风险平价
- 夏普率最大化
- 指标最大化

约束条件

- 个股头寸约束
- 风格约束
- 换手率约束
- 行业约束
- 基准成分股占比
- 追踪误差约束

选股指标

- 预期收益
- 风险贡献
- 风险收益比
- 因子得分

辅助功能

- 股票池筛选
- 股票优先级设置
- 软约束/硬约束设置
- 整合多套行业分类
 - ① 申万一级
 - ② 中信一级
 - ③ 申万一级(金融细分)

整合米筐多因子风险模型

米筐多因子风险模型结合中国市场的涨跌停和停牌制度，对特异风险模型增加了新股、复牌股、停牌股等优化处理，有效提升了风险预测精度和权重优化效果。

10个风格因子

- 贝塔
- 动量
- 规模
- 盈利率
- 杠杆率
- 流动性
- 残余波动率
- 账面市值比
- 非线性市值
- 成长性

28个行业因子

- 采掘
- 化工
- 钢铁
- 电子
- 传媒
- 通信
- 银行
- 综合
- 汽车
- 房地产
- 计算机
- 农林牧渔
- 非银金融
- 有色金属
- 家用电器
- 食品饮料
- 纺织服装
- 轻工制造
- 医药生物
- 公用事业
- 机械设备
- 交通运输
- 商业贸易
- 休闲服务
- 建筑材料
- 建筑装饰
- 电气设备
- 国防军工

1个市场联动因子

- 市场联动因子

使用场景

使用场景	推荐方案	范例
通过有效阿尔法因子获取超额收益，同时希望控制风格、行业、个股特殊风险	<div>· 指标最大化</div> <div>· 风格/行业约束</div> <div>· 个股约束</div>	<pre>import rqdatac from rqoptimizer import * from rqdatac import * rqdatac.init() ### 净利润增长率指数增强策略 def generate_stock_pool(date, indicator_series, stock_number): industry_classification = rqdatac.zx_instrument_industry(indicator_series) index_weight = rqdatac.index_weights('000300.XSHG', date) # 优先选入沪深300成分股中权重大于3%的股票 prioritized_stock_pool = index_weight[index_weight >= 0.03].index prioritized_stock_industry = industry_classification.loc[prioritized_stock_pool] remaining_indicator_series = indicator_series.drop(prioritized_stock_pool) selected_stock = prioritized_stock_pool for i in list(industry_classification.unique()): # 除优先选入股票外，在每个行业选取指标得分最高的股票，使得每一个行业股票总数 industry_prioritized_stock = prioritized_stock_industry[prioritized_stock_industry.index == i] industry_stocks = industry_classification[industry_classification.index == i] industry_selected_stock = remaining_indicator_series.loc[industry_selected_stock == i] selected_stock = selected_stock + industry_selected_stock return selected_stock bounds = {'*': (0, 0.05)} date = '2014-07-16' # 优化日期 # Wildcard的exclude列表为空，即对所有风格/行业设置相同的约束，其中使用中信行业 cons = [WildcardIndustryConstraint(lower_limit=-0.01, upper_limit=0.01, WildcardStyleConstraint(lower_limit=-0.3, upper_limit=0.3, relat] # 获取前一交易日中证800成分股的净利润增长率（TTM） previous_date = rqdatac.get_previous_trading_date(date) index_component = rqdatac.index_components('000906.XSHG', previous_date) indicator_series = rqdatac.get_factor(index_component, 'net_profit_growth', previous_date) selected_stock = generate_stock_pool(previous_date, indicator_series, stock_number) # 个股指标得分范围调整至0.1-1.1，避免权重过分集中于部分指标得分较大的个股 adjusted_series = ((indicator_series.loc[selected_stock] - indicator_series.min()) / (indicator_series.loc[selected_stock].max() - indicator_series.min())) * 1.1 portfolio_weight = portfolio_optimize(selected_stock, date, bnds=bounds, indicator=adjusted_series)</pre>
通过主观选股方式得到股票列表，希望通过优化器控制投资组合风险	<div>· 方差最小化</div> <div>· 风格/行业中性约束</div> <div>· 个股约束</div>	
有投资白名单，没有选股因子，希望实现风格/行业增强的 Smart beta 类型策略	<div>· 选股 API</div> <div>· 风格偏离最小化（因子优先级设置）</div> <div>· 行业约束</div> <div>· 个股约束</div>	
指数增强型策略，希望通过优化器控制追踪误差	<div>· 追踪误差最小化</div> <div>· 风格/行业中性约束</div> <div>· 个股约束</div> <div>· 基准成分股占比约束</div>	
市场处于下行阶段，希望通过优化器优化投资组合风险结构，提高风险分散程度，降低业绩回撤	<div>· 方差或追踪误差最小化</div> <div>· 风格/行业中性约束</div>	

— Part Four ▶▶▶

RQAlpha-Plus回测引擎

支持多品种、多种回测频率

品种	回测频率	范例
股票	日、分钟、tick	<pre>from rqalpha_plus.apis import * from datetime import date # 在这个方法中编写任何的初始化逻辑。context对象将会在你的算法策略的任何方法之间做传递。 #初始化，把标的证券代码定义为变量存储在context中 def init(context): # 标的证券为平安证券 context.s1 = "000001.XSHE" update_universe(context.s1) logger.info("RunInfo: {}".format(context.run_info)) context.counter = 1 context.fired = False def before_trading(context): #print("before") pass def handle_tick(context,tick): # 涨幅计算 change_rate = (tick.last - tick.prev_close) / tick.prev_close # 卖出 if context.fired: # 当前跌幅大于-0.5%且股票可卖的时候，卖出10000股 if change_rate < -0.005 and context.portfolio.positions[context.s1].sellable!=0: print('卖出') order_shares(context.s1, -10000) context.fired = False # 买入 if not context.fired: # 当前涨幅大于1%的时候，建仓，买入10000股 if change_rate>0.01: print('买入') order_shares(context.s1, 10000)</pre>
商品、股指、国债期货	日、分钟、tick	
期权	日、分钟、tick	
可转债	日、分钟、tick	
指数	日、分钟、tick	
交易所债券、银行间债券	日级别	
场内基金	日、分钟、tick	
场外公募基金	日级别	

贴近实战

充分考虑市场事件，提供实战级别的回测。

五类撮合模型

- 当前K线收盘价
- 下一K线开盘价
- 最新价
- 己方最优报盘价格
- 对手方最优报盘价格

两类滑点模型

- 固定比例滑点模型：
成交价格按照一定比例受影响
- 固定价格滑点模型：
成交价格按最小价格变动单位受影响

股票

- T+1
- 分红拆分
- 成交量限制

基金

- 申购赎回
- 分红
-

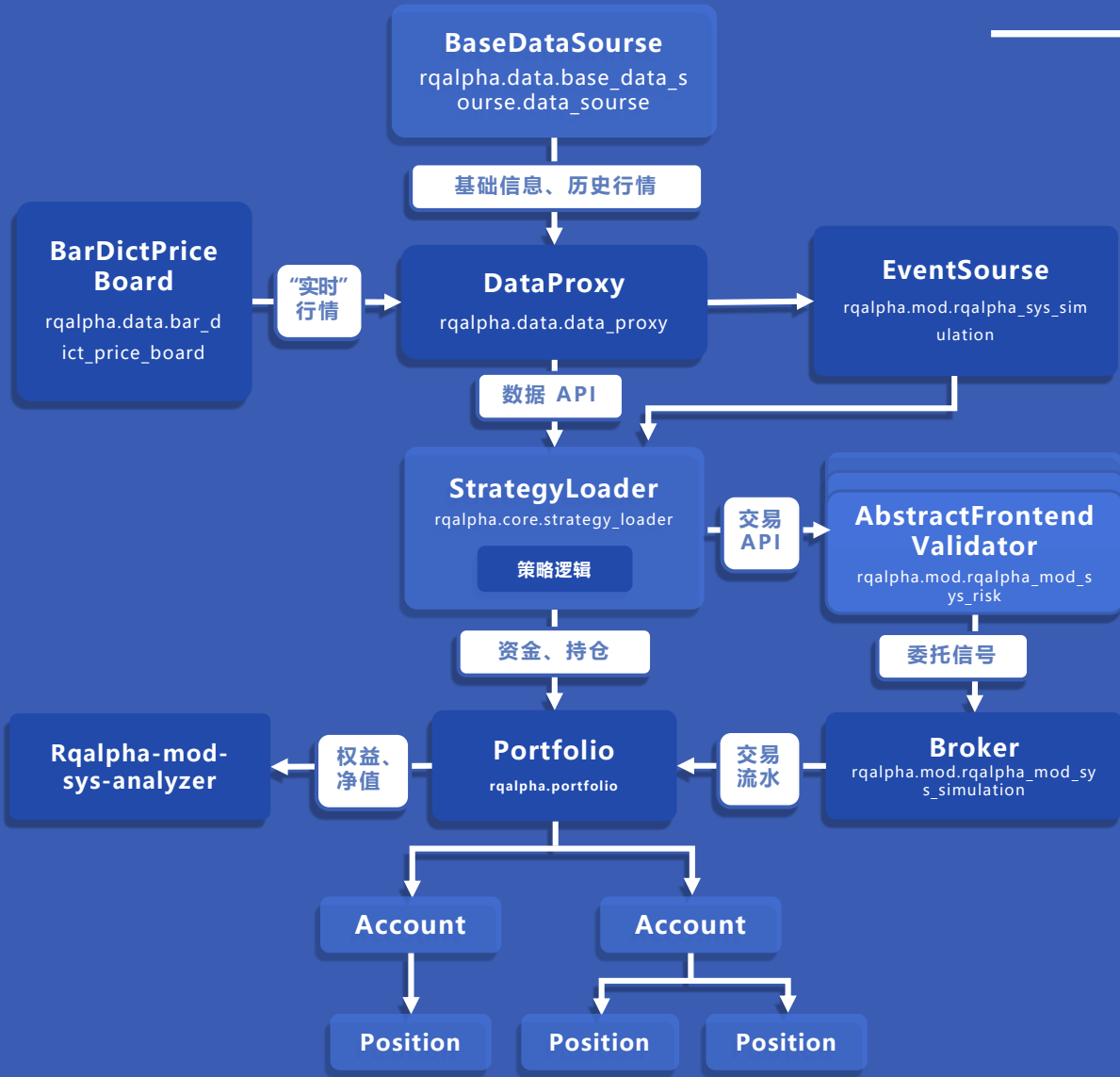
期权

- 行权操作
- 行权滑点
-

债券

- 回购
- 逆回购
-

高度可扩展性



引入Mod机制，具有高度可扩展性，方便用户二次开发。

◀ RQAlpha通过Mod机制实现扩展 ▼ RQAlpha接口代码示例

```
145 class AbstractStrategyLoader(with_metaclass(abc.ABCMeta)):
146     """
147     策略加载器，其主要作用是加载策略，并将策略运行所需的环境传递给策略执行代码。
148     在扩展模块中，可以通过调用 ``env.set_strategy_loader`` 来替换默认的策略加载器。
149     """
150     @abc.abstractmethod
151     def load(self, scope):
152         """
153         [Required]
154         Load 函数负责组装策略代码和策略代码所在的环境，并输出最终组装好的可执行域。
155         :param dict scope: 策略代码运行环境。在传入时，包含了所有基础API。
156         通过在 scope 中添加函数可以实现自定义API；通过覆盖 scope 中相应的函数，可以覆盖API。
157         :return: scope, 其中应包含策略相应函数，如 ``init``, ``before_trading`` 等
158         """
159         raise NotImplementedError
160
161 class AbstractEventSource(with_metaclass(abc.ABCMeta)):
162     """
163     事件源接口。RQAlpha 从此对象中获取事件，驱动整个事件循环。
164     在扩展模块中，可以通过调用 ``env.set_event_source`` 来替换默认的事件源。
165     """
166     @abc.abstractmethod
167     def events(self, start_date, end_date, frequency):
168         """
169         [Required]
170         扩展 EventSource 必须实现 events 函数。
171         events 是一个 event generator，在相应事件的时候需要以如下格式来传递事件
172         .. code-block:: python
173             yield trading_datetime, calendar_datetime, EventEnum
174         """
```

认可度高

代码开源，用户基础大，
经过大量机构验证。

https://github.com/topics/rqalpha

Explore Topics Trending Collections Events GitHub Sponsors

rqalpha

Here are 7 public repositories matching this topic...

Language: All ▾

ricequant / rqalpha

☆ Star 3.9k

<> Code ! Issues 🔗 Pull requests

A extendable, replaceable Python algorithmic backtest && trading framework supporting multiple securities

python finance trading stock quant futures ta-lib backtest

ricequant rqalpha

Updated 4 hours ago ● Python

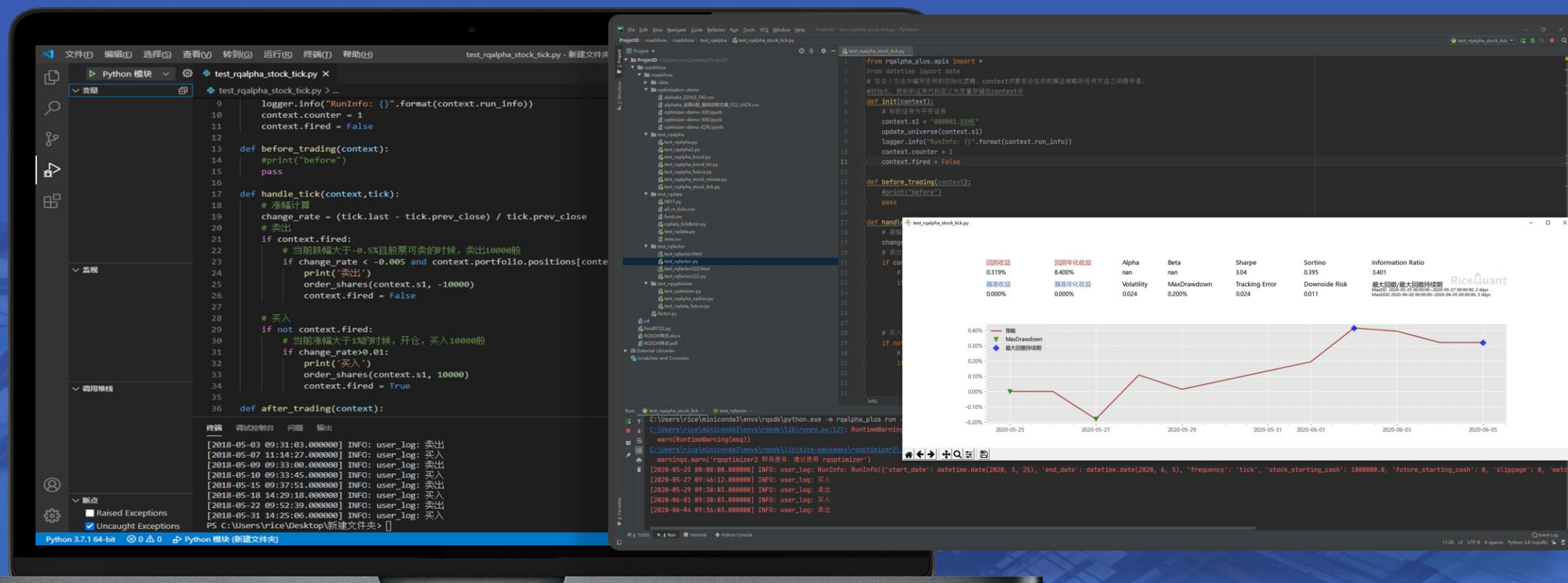
Page 22

策略本地化

RQSDK可与本地开发环境集成，用户策略完全本地化运行，可引用丰富的第三方库。

▼ RQSDK与VSCode集成示例

▼ RQSDK与PyCharm集成示例



— RQSDK优势 ▶▶▶

- 功能丰富：支持数据、回测、因子投研和组合优化等多种功能，覆盖量化投研全流程。
- 品种齐全：支持中国市场几乎所有场内金融工具，以及公募基金和银行间债券。
- 稳定高效：产品经过多年迭代，以多种形态被众多客户使用，稳定高效。
- 高度解耦：不同产品之间互相协同，又可独立协作，用户可按需购买。

内容



米筐科技Smartbeta策略研究：指数增强型股息率策略（节选）

红利选股是一类重要的投资策略。能够进行现金分红的上市公司通常基本面良好，账面有充足的现金流；而分红再投资则能为投资.....



我国目标日期基金发展浅论

探讨生命周期配置在中国可能适用的发展模式及自身的几点思考。



贯穿生命周期的资产配置：目标日期基金

以全球养老金系统发展最为成熟的美国为例，探讨目标日期基金的构建方法。



投资组合优化风险度量工具比较

风险度量工具的选择在投资组合优化中起着基础和决定性的作用。



行业基本面选股：基本面分析的量化操作指南

行业基本面选股是量化投资和基本面分析的有机结合。不同于传统的多因子选股，行业基本面选股更关注公司经营的本质，从基.....



RQAMS米筐资产管理系统正式发布！| 免费试用开启

RQAMS的发布作为一个新的起点，米筐将继续以满足资管业务和财富管理的核心需求为宗旨，不断迭代完善产品。



扫码关注米筐科技公众号

感谢观看

联系我们

深圳米筐科技有限公司
WWW.RICEQUANT.COM

合作邮箱: BD@ricequant.com 联系电话: 0755-2267-6337
公司地址: 深圳市南山区高新南七道深圳国家工程实验大楼A栋304