

$$\frac{d}{dx} \cosh x = \sinh x$$

$$\csc(-\theta) = -\csc(\theta)$$



∞

$$\frac{f(x)}{g(x)} = \lim_{x \rightarrow a} \frac{f'(x)}{g'(x)} = \lim_{x \rightarrow a} \frac{f'(x)}{g'(x)} g(x) + \frac{d}{dx} f(x) g(x)$$

$$\frac{d}{dx} (f(x) \lim_{x \rightarrow a} \frac{f(x)}{g(x)}) = \lim_{x \rightarrow a} \frac{f'(x)}{g'(x)} g(x)$$

$$\frac{n+1}{X} \quad a^2 - b^2 = (a + b)(a - b)$$

$$\sum_{k=1}^n k^2 = \frac{n(n+1)(2n+1)}{6} \quad X \quad a^2$$

$$F = ma$$

$$F =$$

χ

$$\sin 2\theta = 2 \sin \theta \cos \theta$$

Ω

$$\frac{d}{dx} \arccsc x = \frac{d}{dx} \csc^{-1} x = \frac{-1}{|x| \sqrt{x^2 - 1}}$$

$$\sin^2 \theta + \cos^2 \theta = 1$$

Made in Ural

λ

$$\sin^2 \theta +$$

$$\cot(-\theta) = -\cot(\theta)$$

RUSQUANT



Торгуй стратегии, а не эмоции

Rusquant bot

Хакатон Finam Trade API

Москва, Сколково, 20.05.2023

Команда

Vyacheslav Arbuzov, PhD

- ✓ Fintech entrepreneur, Trader
- ✓ Ex-assistant professor, Perm State University, Russia
- ✓ Ex-quant, hedge fund, Moscow, Russia

arbuzov1989@gmail.com
t.me/arbuzovv

chatGPT 

- ✓ Junior developer
- ✓ Marketing
- ✓ Copywriter
- ✓ Docs



Задачи

Создание торгового робота удовлетворяющего следующим критериям:

- Количественная торговля
 - Алгоритмическая торговля
 - Минимальное время на мониторинг рынка
 - Конец дня, момент принятие решений
 - Российский рынок
 - Торговля фьючерсами
 - Минимальная комиссия, дешевое фондирование
 - Вселенная: 40 акций на MOEX для которых есть фьючерсы
 - Ансамбль моделей принимающих решения
 - Диверсификация по инструментам и стратегиям
 - Легкость размещения, администрирования и оплаты инфраструктуры для робота
 - Возможность продать побочный продукт при создании робота
 - Хостинг робота на отечественной облачной платформе
-
- Точно не HFT
 - Точно не Discretionary
 - Точно не Intraday trading



Альфа стратегии

Альфа (или альфа-стратегия) - это комбинация математической формулы, компьютерного кода и параметров, которая при добавлении к ним исторических данных способна прогнозировать будущие движения различных финансовых инструментов

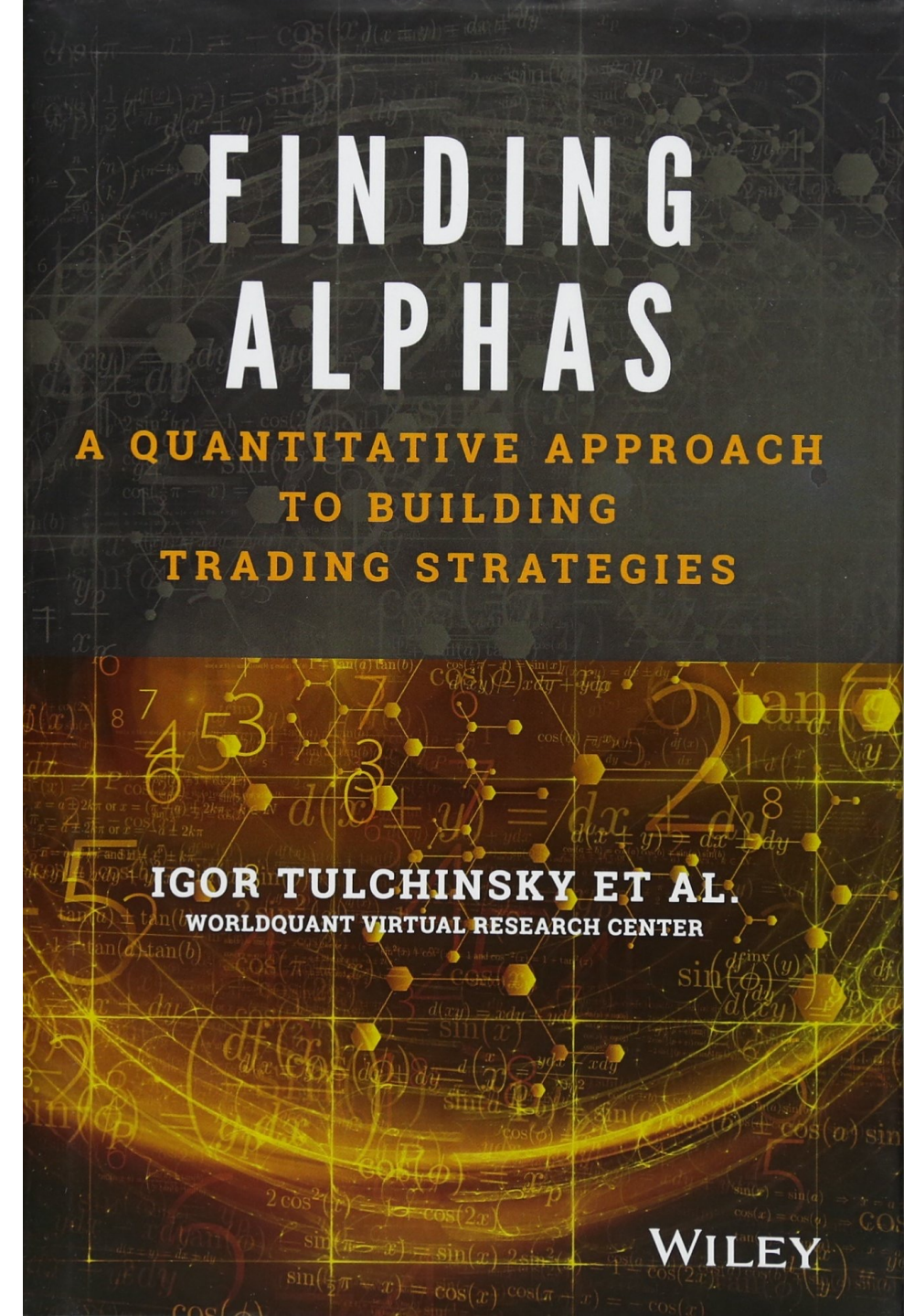
$$High > \overline{High_{20}} ? 1:0$$

$$sign[(V_t - V_{t-1}) * (Close_t - Close_{t-1})]$$

$$\frac{Close - Open}{High - Low + 0.001}$$

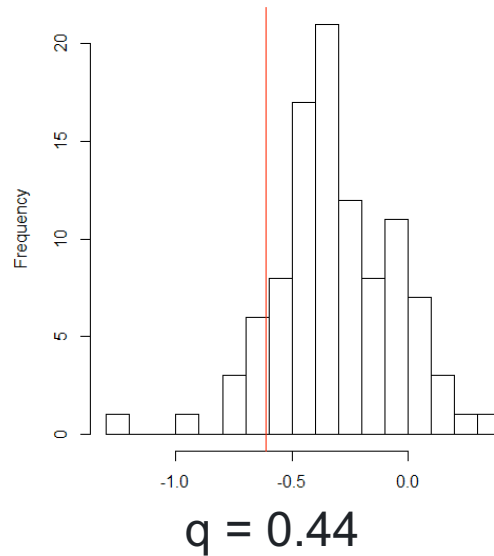
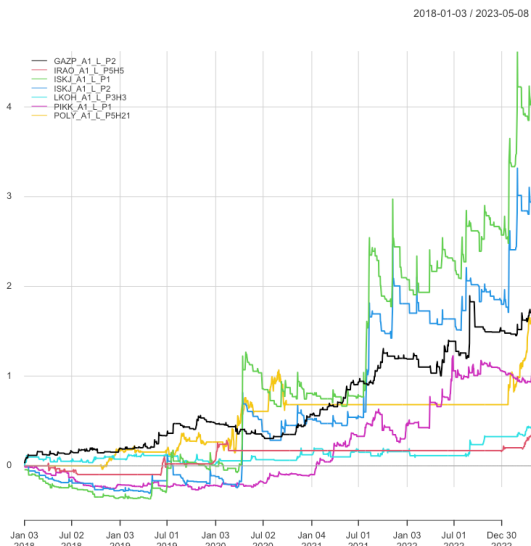
Дополнительная литература

- Kakushadze, Z. (2016) 101 Formulaic Alphas. Wilmott Magazine 2016(84): 72-80.



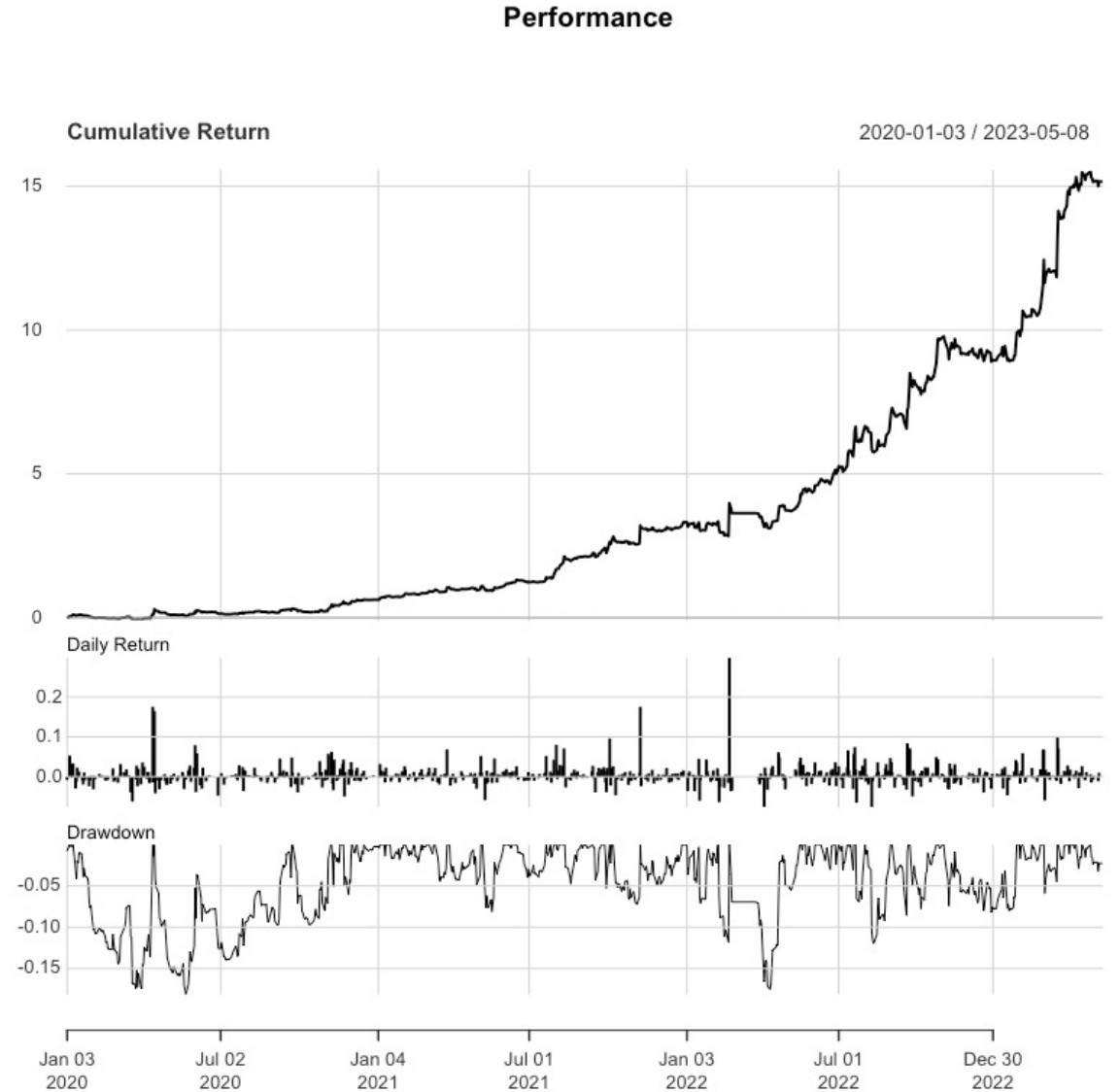
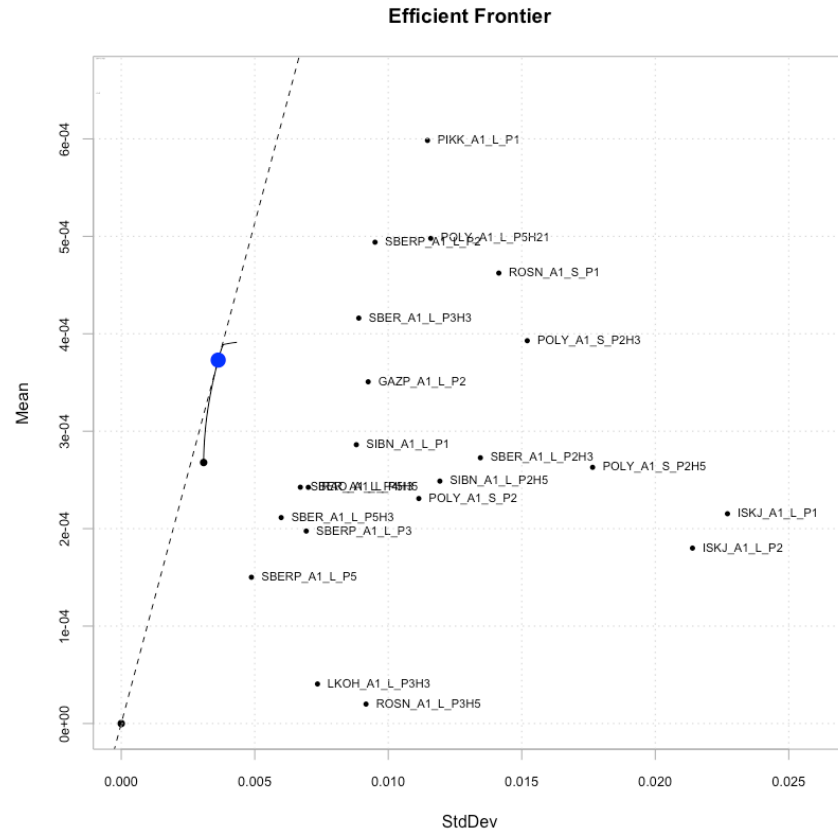
Бектестинг альфа стратегий

- Доходность и коэффициент Шарпа / Кальмара самые используемые метрики для выбора стратегии
- Выбрать лучшие по Шарпу и торговать?
- Бутстреп приростов цены и генерация фейковых траекторий
- Оценка альфы на фейковых траекториях
- Сравнение шарпа/кальмара на фейковых и реальных данных
- Метрика – квантиль эмпирического распределение



Портфель альфа стратегий

- Собираем все альфы выбранные на предыдущем этапе в портфель альфа стратегий
- Добавляем ограничение на альфу
- Добавляем ограничение на позицию в инструменте
- Максимизируем Шарп портфеля



RUSQUANT API

Почему API:


- Тысячи временных рядов
- Сложность в хранении и управлении
- Обработка и вычисления в конце дня
- Дальнейшая масштабируемость в добавлении новых рынков и новых альф
- Features-as-a-Service (FaaS)
- Возможность продавать доступ к API

Реализованная API:

- Основа OpenAPI 3.0
- Доступ API используя API – ключ (можно получить тестовый доступ на rusquant.ru)
- Инфраструктура в Yandex Cloud
- Расчет альф в конце дня
- Метаинформация для выбора торгуемых альф
- Реализовано 45 разнообразных альф



swagger.rusquant.ru

 Swagger
Supported by SMARTBEAR

<https://swagger.rusquant.ru/api-rusquant.json>

Explore

Rusquant API ^{1.0} ^{OAS3}

<https://swagger.rusquant.ru/api-rusquant.json>

API for accessing alpha data.

Servers

https://api.rusquant.ru - Official API of Rusquant

default

GET

/performance

Get performance of data

GET

/alpha

Get alpha for a symbol

Get the available data for a given symbol and alpha

Parameters

Try it out

Реализация торгового робота на R

Почему R:

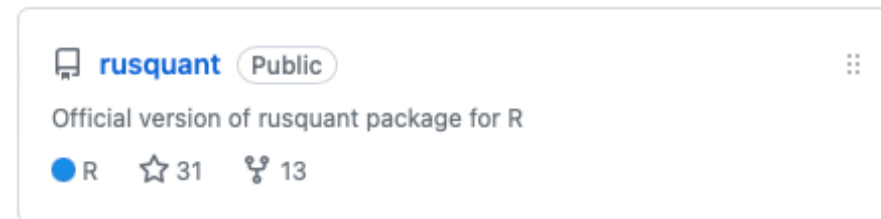
- Язык R — язык общения с данными
- Прародитель современных библиотек на Python
- Основной язык для академических исследований
- Огромное количество библиотек для количественных финансов
- Используется во многих фондах как основной язык исследования
- Входит в топ-10 языков программирования на 2023 год (PYPL Index)
- Удобный инструментарий RStudio
- Большое комьюнити квантов (R in Finance)

Библиотека **rusquant** для R:

- Реализована поддержка торговли через Finam Trade API (методы REST API)
- Реализована поддержка Rusquant API
- Открытые источники данных
- Библиотека загружена на CRAN (для возможности использования на Yandex Cloud)

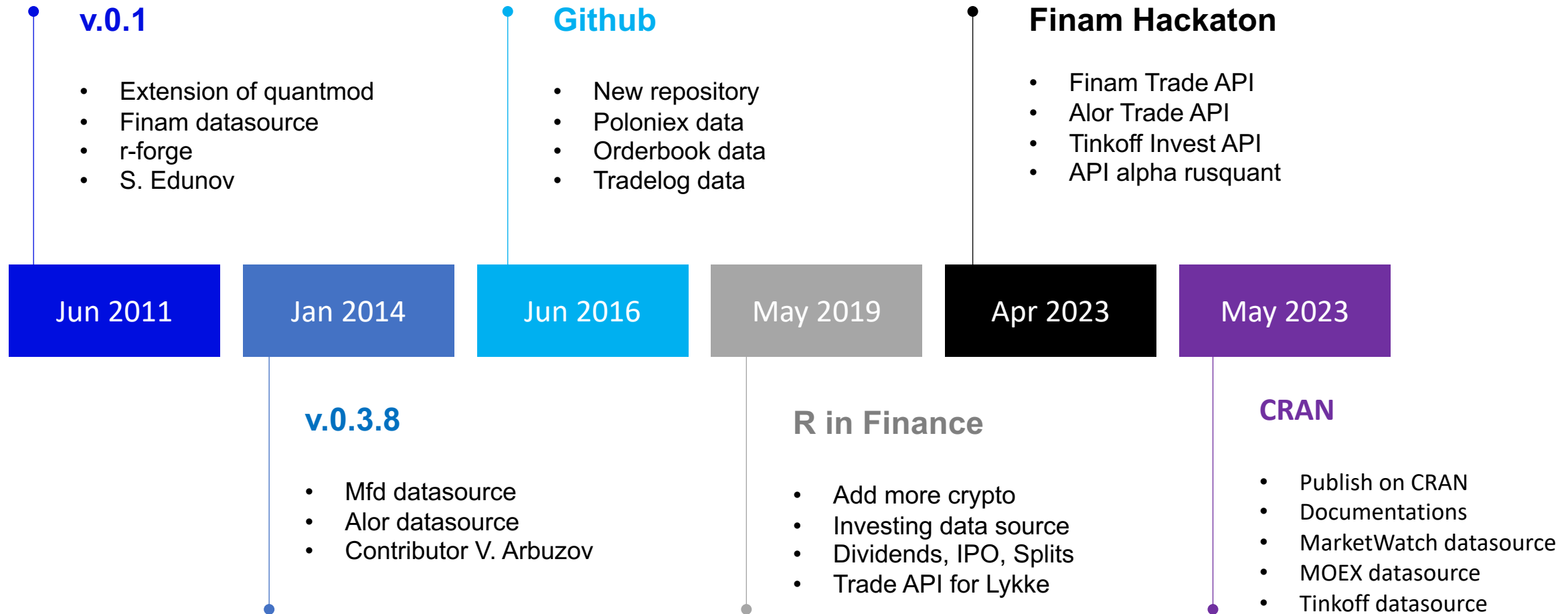


Олег Мубаракшин, Joshua Ulrich (quantmod), Вячеслав Арбузов
R in Finance conference, Chicago, USA, 2014



CRAN 1.0.1 downloads 140/month CRAN 8 days ago

Развитие библиотеки Rusquant



Rusquant сейчас это

Поддержка источников данных:



Поддержка торгового API:



Команды в библиотеке rusquant

Получить позиции в портфеле

>

Получить справочник инструментов

> |

Выставить заявку

> |

Список заявок

> |

Запросить количество денег

>

Получить цены инструмента

> |

Отменить заявку

> |

Работа с API rusquant

> |

Код торгового робота Rusquant bot

Файл quant.R – файл для выбора торгуемых альф и их весов.

1. Выбор анализируемых альф

```
# get list of alpha performance
alpha_performance = getSymbolList(src='Rusquant', api.key=rusquant_token)
alpha_universe = alpha_performance[sharpe_3y>0.8 & sharpe_1y>0.8 &
  sharpe_1y<2 & sharpe_1m<3 &
  ret_1m>0 & ret_1m<0.4 & q<0.6]
```

2. Загрузка альф и открытых ценовых данных

```
for(i in 1:nrow(alpha_universe))
{
  print(alpha_universe[i])
  alpha_i = getSymbols.Rusquant(alpha_universe$symbol[i],field = alpha_universe$alpha[i],
    from = '2010-01-01',to=Sys.Date(),api.key = rusquant_token)
  price = try(getSymbols.Finam(alpha_universe$symbol[i],from = '2010-01-01',
    auto.assign = F),silent = TRUE)
  ret = ROC(price[,4])
  future_ret = lag(ret,-1)
  alpha_signal = xts(alpha_i$signal,order.by = as.Date(alpha_i$date))
  alpha_signal_ret = merge(alpha_signal,future_ret)
  alpha_signal_ret$pnl = alpha_signal_ret[,1]*alpha_signal_ret[,2]
  alpha_signal_ret = na.omit(alpha_signal_ret)[,3]
  if(i==1) res = alpha_signal_ret
  if(i!=1) res = merge(res,alpha_signal_ret)
}
```

3. Оптимизация портфеля

```
# create virtual object portfolio
pf = portfolio.spec(alpha_universe$symbol_alpha)
# add add.constraints
pf = add.constraint(portfolio=pf, type="full_investment")
pf = add.constraint(portfolio=pf, type="long_only")
pf = add.objective(portfolio=pf, type="return", name="mean")
pf = add.objective(portfolio=pf, type="risk", name="StdDev")
pf = add.constraint(portfolio=pf, type="box", min=0.01, max=0.1)
pf <- add.constraint(portfolio=pf, type="group",
  groups=group_list,
  group_min = rep(0,length(group_list)),
  group_max = rep(0.2,length(group_list)))
# run optimizer
opt_portf <- optimize.portfolio(R=na.omit(res[time_period]), portfolio=pf,
  optimize_method="ROI",
  maxSR=TRUE, trace=TRUE)
```

Файл trade.R – файл для ежедневной торговли

а. Вычисление теоретических позиций на текущий день

```
# get signals for current date
for(i in 1:nrow(alpha_universe))
{
  alpha_i = getSymbols.Rusquant(alpha_universe$symbol[i],field = alpha_universe$alpha[i],
    from = '2010-01-01',to=Sys.Date(),api.key = rusquant_token)
  if(i==1) signals = data.table(alpha_universe$symbol[i],
    alpha_universe$alpha[i],tail(alpha_i,1)$signal)
  if(i!=1) signals = rbind(signals,data.table(alpha_universe$symbol[i],
    alpha_universe$alpha[i],tail(alpha_i,1)$signal))
}
signals$w = signals$V3*w_opt
position_per_symbol = signals[,sum(w,na.rm = T),by='V1']
```

б. Текущий портфель

```
# actual portfolio
finam_portfolio = getPortfolio(src = 'Finam',
  api.key = finam_token,clientId = finam_account)
capital = finam_portfolio$money$balance * leverage # get from Portfolio
position_per_symbol$money = capital*position_per_symbol$w
```

с. Разница между теоретической и практической позицией

```
# diff between theoretical and actual portfolio
change_portfolio = merge(trade_positions,
  finam_portfolio$positions[c('securityCode','balance')],
  by.x = 'futures_code',by.y = 'securityCode',all = T)
change_portfolio[is.na(size)]$size = 0
change_portfolio[is.na(balance)]$balance = 0
change_portfolio$trade_size = change_portfolio$size - change_portfolio$balance
```

д. Отправка заявок

```
#change positions
for(i in 1:nrow(change_portfolio))
{
  trade_symbol = change_portfolio$futures_code[i]
  last_price = try(tail((getSymbols.Finam(trade_symbol,period = '1min',
    from=Sys.Date()-2)[,4,1],silent = T)
  size_order = abs(as.numeric(change_portfolio$trade_size[i]))
  trade_side = ifelse(change_portfolio$trade_size[i]>0,'Buy','Sell')
  myorder = placeOrder(src = 'finam',
    symbol = trade_symbol,
    board = 'FUT',
    action = trade_side,
    totalQuantity = size_order,
    lmtPrice = as.numeric(last_price),
    api.key = finam_token,
    clientId = finam_account))}
```

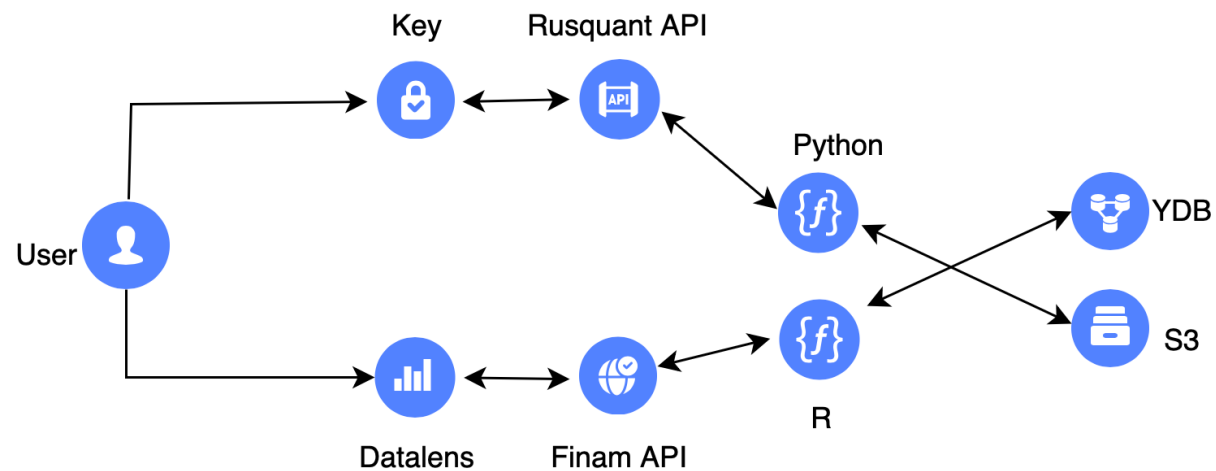
Архитектура проекта

Преимущества:

- Простота кода итогового робота
- Почти бесплатный хостинг робота
- Надежность и бесперебойность
- Микросервисный подход в создании торгового робота

Ограничения:

- Отсутствует интерактивность присущая терминалам



Исследования,
вычисления, доставка

RUSQUANT API

Удобный доступ

RUSQUANT PACKAGE

Вычисление позиций,
отправка заявок

RUSQUANT BOT

Железо и хостинг

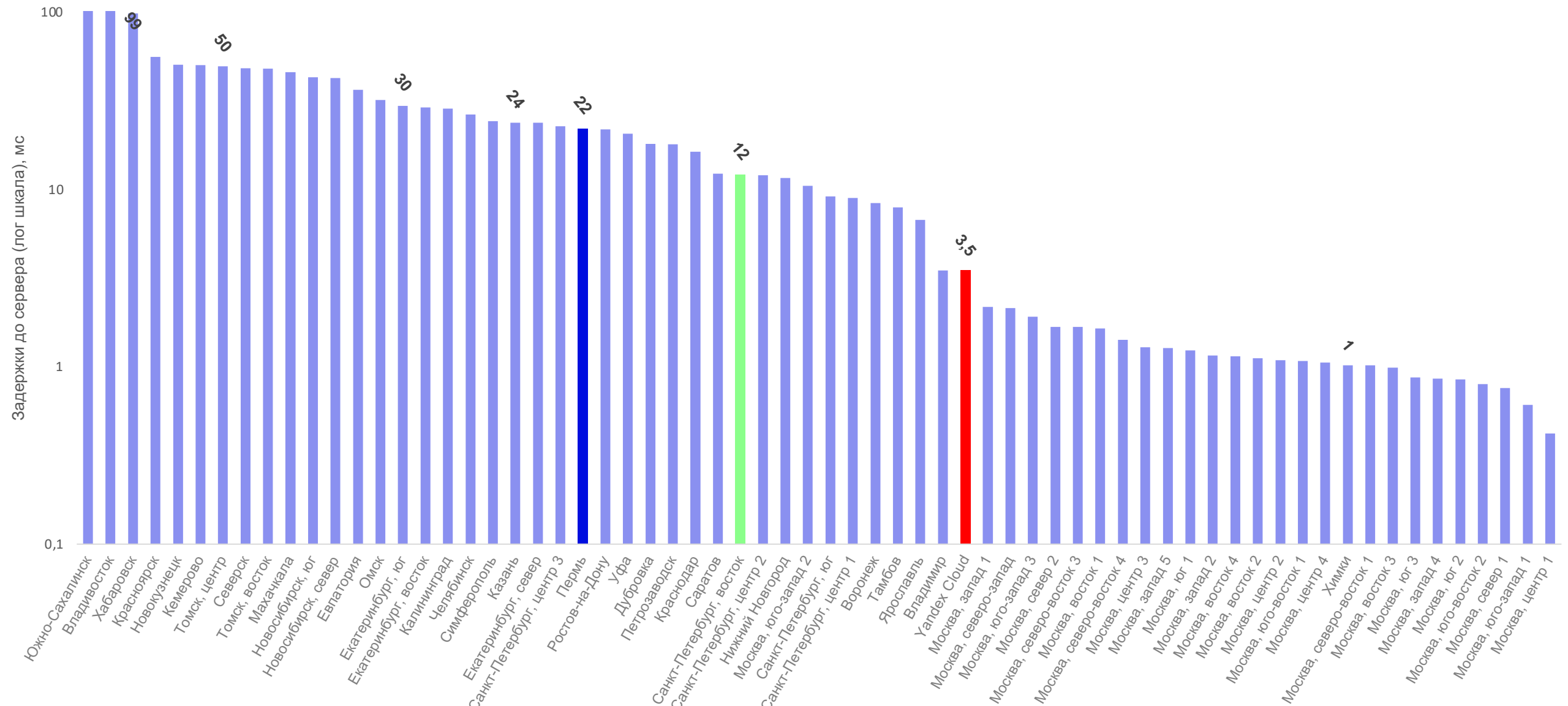
YANDEX CLOUD

Связь с брокером

FINAM TRADE API

EXCHANGE

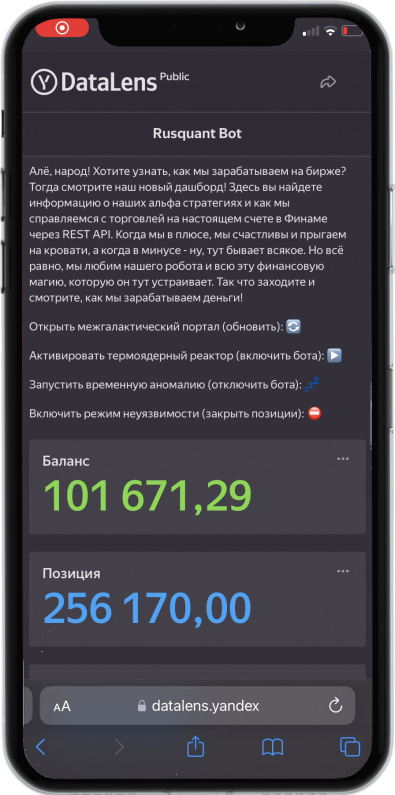
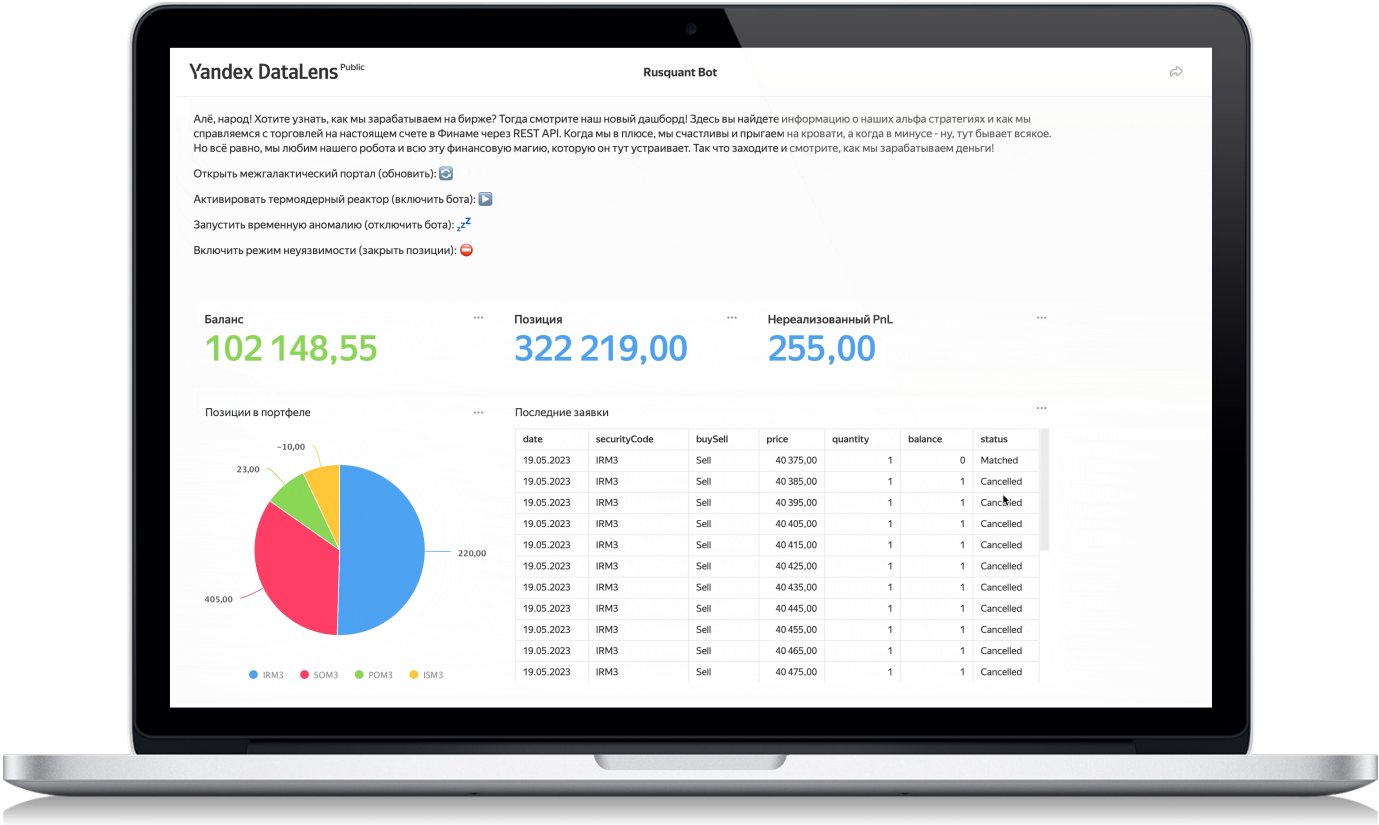
Задержки до сервера Finam



Управление торговым роботом

Вся аналитическая информация собирается с использованием Finam API и размещается на дашборде сервиса Yandex Cloud – Data Lens. Все данные обновляются по заданному расписанию. Управление происходит с помощью кнопок на дашборде.

dataLens.yandex.ru



Демонстрация работы торгового робота

```
1 # install.packages("rusquant") # if not installed
2 library(devtools)
3 library(xts)
4 library(TTR)
5 library(quantmod)
6 library(PortfolioAnalytics)
7 library(PerformanceAnalytics)
8 library(rusquant)
9 library(telegram)
10
11 # set params for your account
12 leverage = 3
13
14 # token data for Finam Hackaton
15 rusquant_token = 'AAHXE1642J'
16
17 ##### Trading execution logic #####
18
19 w_opt = fread('w_opt.csv')
20 alpha_universe = fread('alpha_universe.csv')
```

79:23 Trading execution logic

R Script

Console Terminal Render Background Jobs

R 4.0.5 ~/

```
> change_portfolio
  futures_code symbol      w  money tick_size price size balance trade_size
1:      IRM3  IRAO 0.10000000 30300.987      0 40475   1      2      -1
2:      ISM3  ISKJ 0.02000000  6060.197      0  1136   5      0      5
3:      LKM3  LKOH 0.01000000  3030.099      0  47077   0      0      0
4:      POM3  POLY 0.01184456  3589.017      0  6610   1      1      0
5:      RNM3  ROSN 0.01000000  3030.099      0  41617   0      0      0
6:      SOM3  SIBN 0.08906554 26987.738      0  5437   5      5      0
> |
```

Спасибо за внимание!



@ARBUZOVV



rusquant.ru



arbuzov1989@gmail.com



t.me/arbuzovv



github.com/arbuzovv

Made in Ural