

Ψ

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

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

 Θ ∞

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

$$x^{(n+1)}$$

$$a^2 - b^2 = (a+b)(a-b)$$

$$F = ma$$



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

 Ω χ

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

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

Made in Ural
 λ

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

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

$$=\oint_C E \cdot d\ell$$

$$F =$$

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

$$\sin^2 \theta +$$

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 

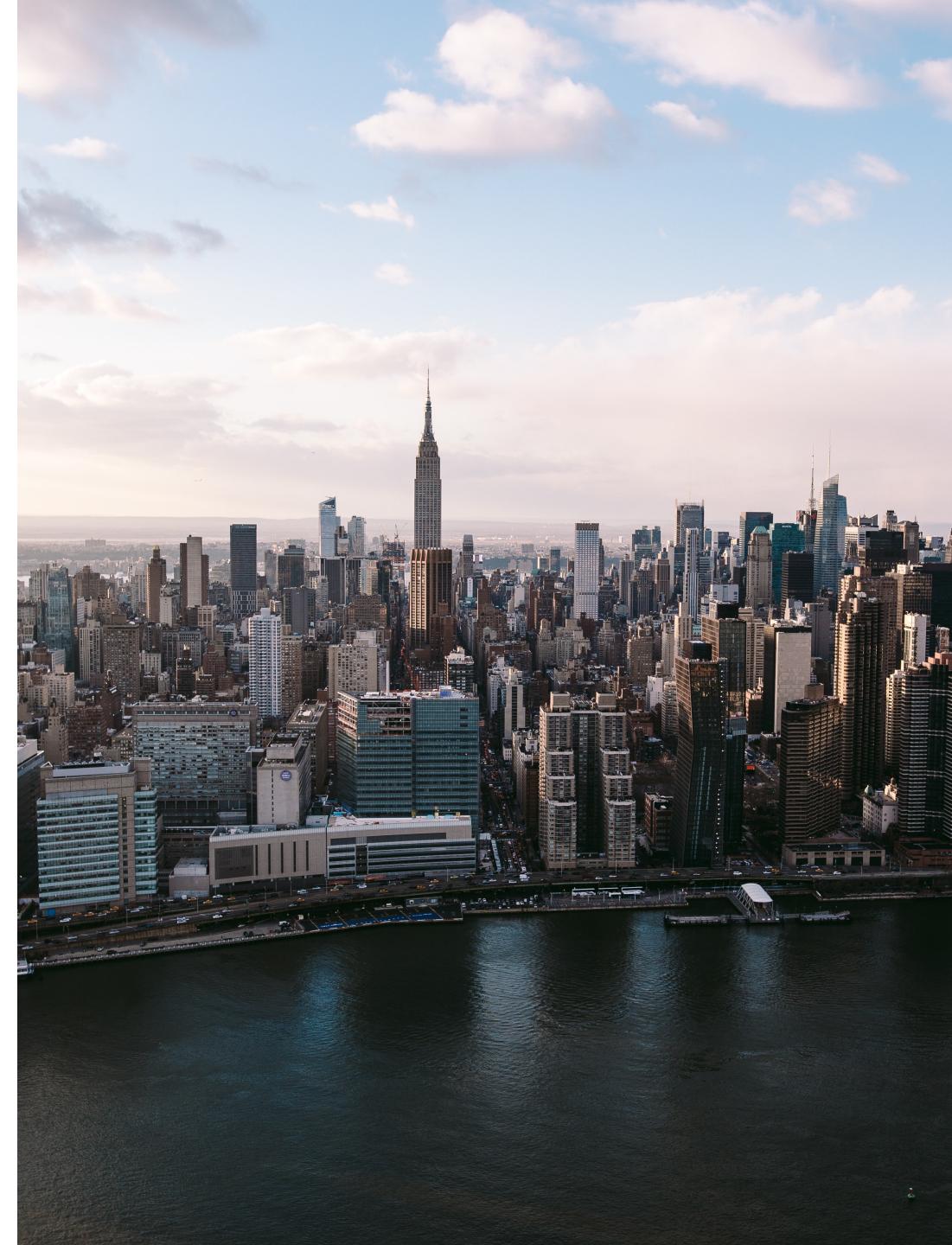
- ✓ Junior developer
- ✓ Marketing
- ✓ Copywriter
- ✓ Docs



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 МОEX 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 различный финансовых инструментов

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

$$\text{sign}[(V_t - V_{t-1}) * (\text{Close}_t - \text{Close}_{t-1})]$$

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

Дополнительная литература

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

FINDING ALPHAS

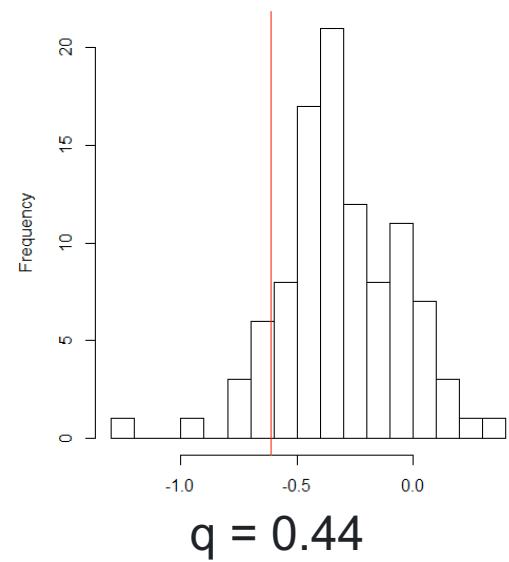
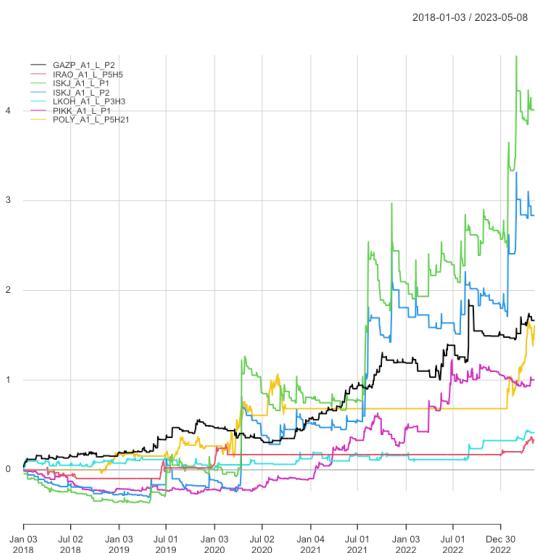
A QUANTITATIVE APPROACH
TO BUILDING
TRADING STRATEGIES

IGOR TULCHINSKY ET AL.
WORLDQUANT VIRTUAL RESEARCH CENTER

WILEY

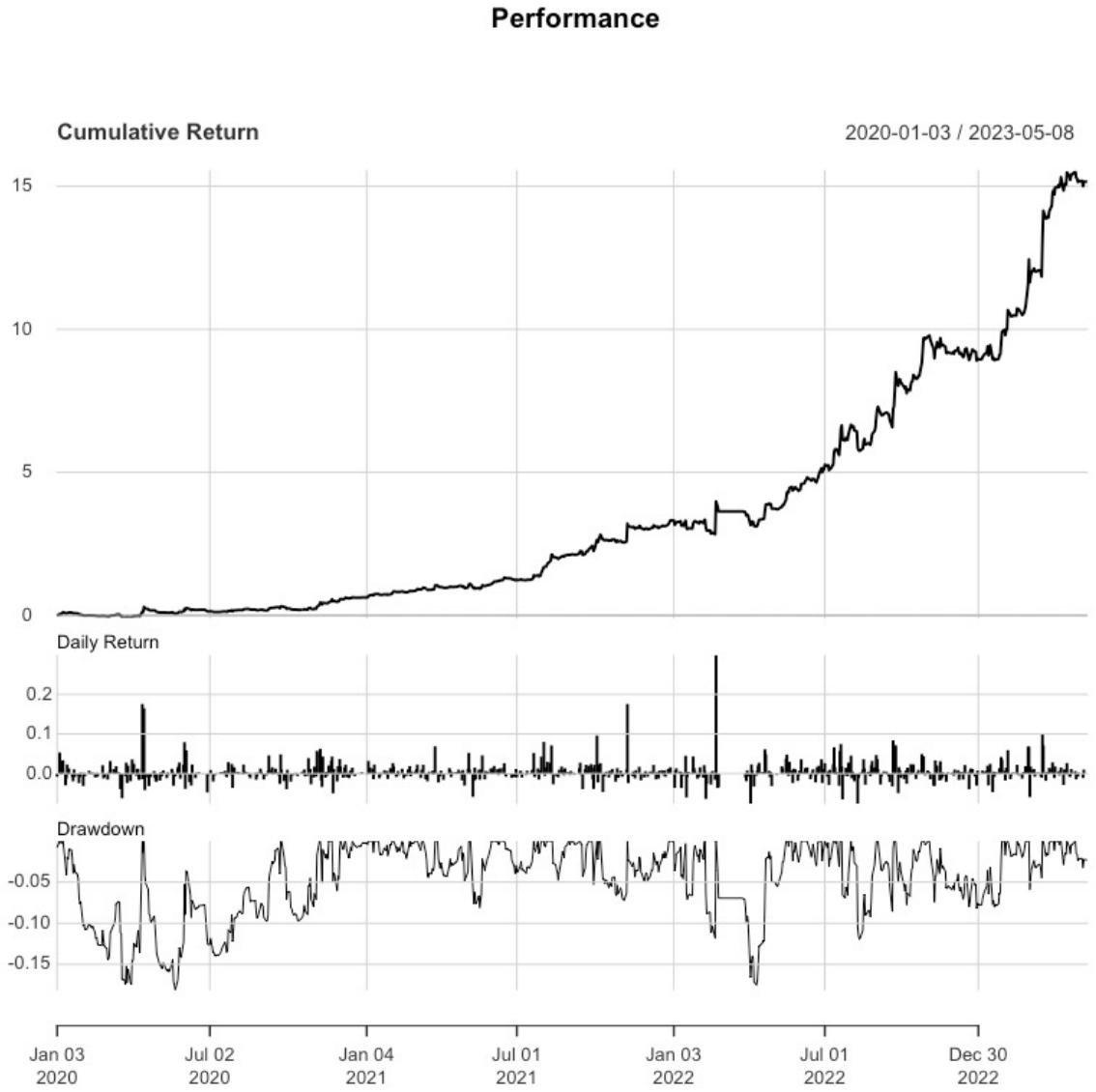
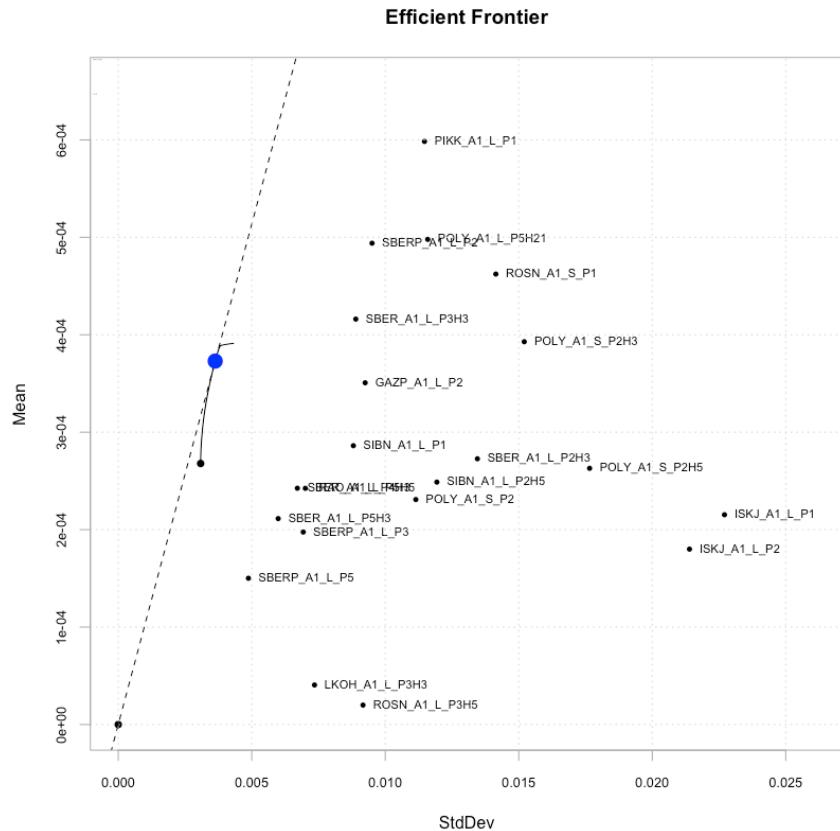
Бектестинг альфа стратегий

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



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

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

Name	Description
------	-------------

symbol * required	The symbol for which to retrieve data
string	
(query)	
<input type="text" value="symbol"/>	

alpha * required	The feature for which to retrieve data
string	
(query)	
<input type="text" value="alpha"/>	

from	The dates to for which to retrieve data
string	
(query)	
<input type="text" value="from"/>	

to	The dates to for which to retrieve data
string	
(query)	
<input type="text" value="to"/>	

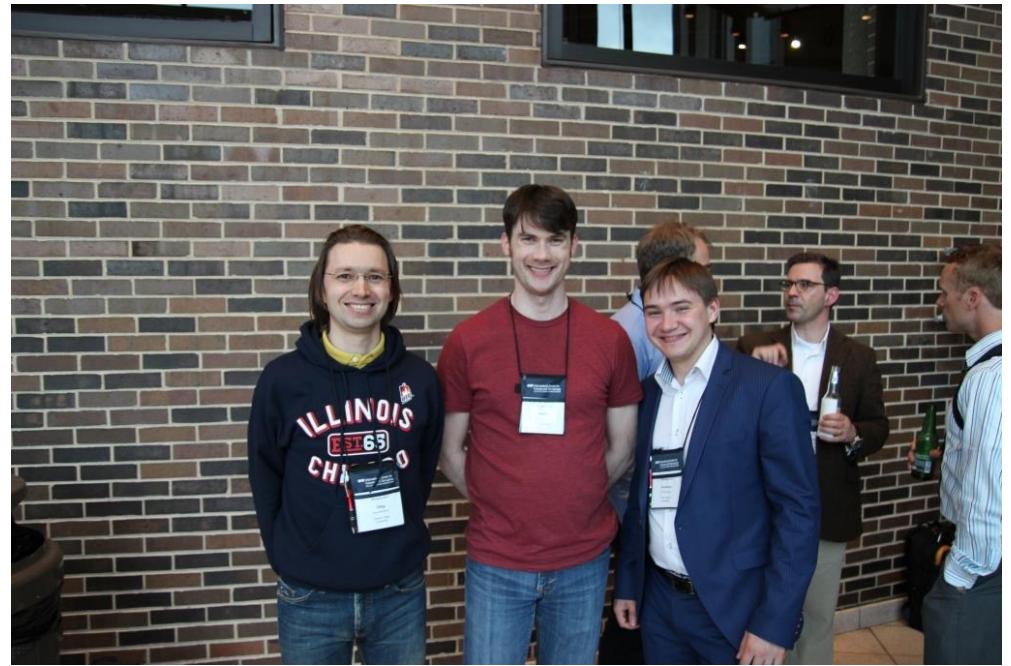
Реализация торгового робота на R

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

 **rusquant** Public

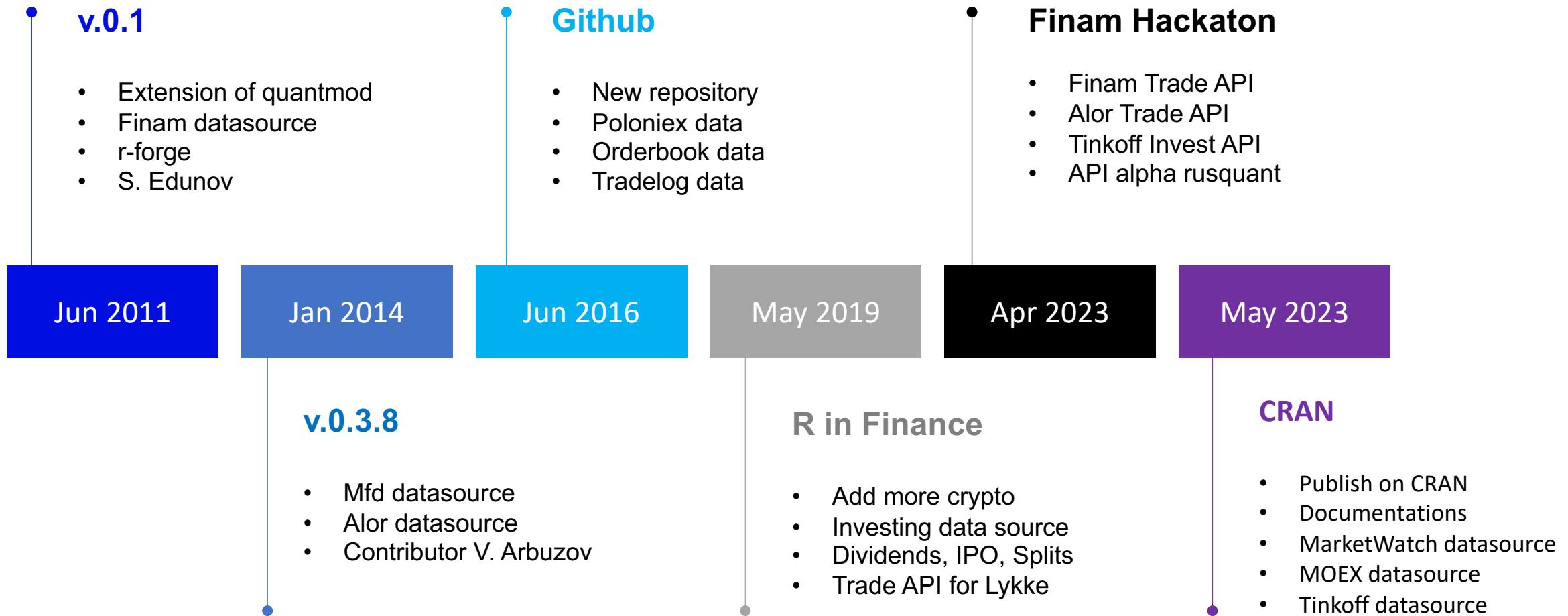
Official version of rusquant package for R

 R  31  13

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

a. 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']
```

b. Текущий портфель

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

c. 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
```

d. 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)}
```

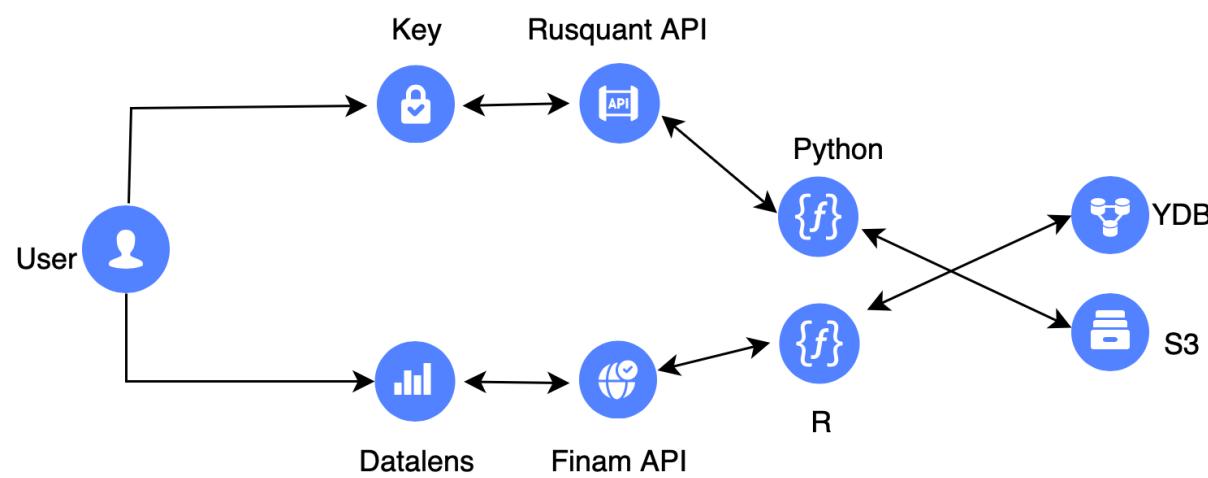
Архитектура проекта

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

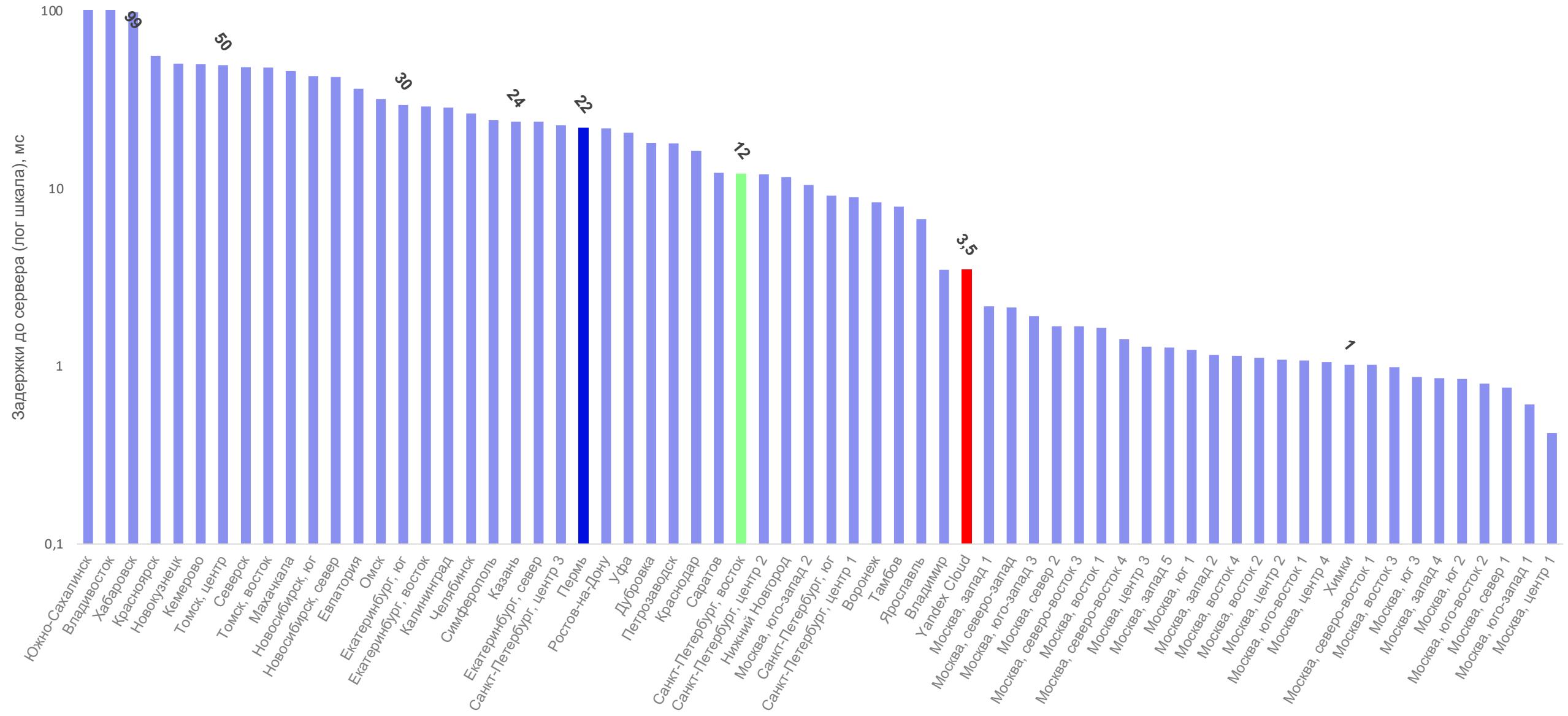
YANDEX CLOUD

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



The tablet screen shows the Yandex DataLens Public dashboard for the 'Rusquant Bot'. Key metrics displayed include:

- Баланс (Balance): 102 148,55
- Позиция (Position): 322 219,00
- Нереализованный PnL (Unrealized PnL): 255,00

Below these metrics is a pie chart titled "Позиции в портфеле" (Positions in portfolio) showing the distribution of assets:

Акция (Security)	Кол-во (Quantity)	Цена (Price)
IRM3	23,00	40 375,00
SOM3	405,00	40 385,00
POM3	10,00	40 395,00
ISM3	220,00	40 405,00

On the right side of the dashboard, there is a table titled "Последние заявки" (Recent orders) showing a list of sell orders for IRM3:

date	securityCode	buySell	price	quantity	balance	status
19.05.2023	IRM3	Sell	40 375,00	1	0	Matched
19.05.2023	IRM3	Sell	40 385,00	1	1	Cancelled
19.05.2023	IRM3	Sell	40 395,00	1	1	Cancelled
19.05.2023	IRM3	Sell	40 405,00	1	1	Cancelled
19.05.2023	IRM3	Sell	40 415,00	1	1	Cancelled
19.05.2023	IRM3	Sell	40 425,00	1	1	Cancelled
19.05.2023	IRM3	Sell	40 435,00	1	1	Cancelled
19.05.2023	IRM3	Sell	40 445,00	1	1	Cancelled
19.05.2023	IRM3	Sell	40 455,00	1	1	Cancelled
19.05.2023	IRM3	Sell	40 465,00	1	1	Cancelled
19.05.2023	IRM3	Sell	40 475,00	1	1	Cancelled

The smartphone screen shows the mobile version of the Yandex DataLens Public dashboard for the 'Rusquant Bot'. The key metrics are identical to the tablet version:

- Баланс (Balance): 101 671,29
- Позиция (Position): 256 170,00

The interface includes a header with the DataLens logo and a footer with navigation icons and the URL dataLens.yandex.

Демонстрация работы торгового робота

The screenshot shows an RStudio interface with an R script file open and an R console window below it.

R Script Content:

```
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
15 # token data for Finam Hackaton
16 rusquant_token = 'AAHXE1642J'
17
18 ##### Trading execution logic #####
19
20 w_opt = fread('w_opt.csv')
21 alpha_universe = fread('alpha_universe.csv')
22
```

R Console Output:

```
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 0.1000000 30300.987      0 40475   1     2    -1
2:     ISM3     ISKJ 0.0200000 6060.197      0 1136    5     0     5
3:     LKM3     LKOH 0.0100000 3030.099      0 47077   0     0     0
4:     POM3     POLY 0.01184456 3589.017      0 6610    1     1     0
5:     RNM3     ROSN 0.0100000 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