

Количественные финансы

Марсель Салихов

Invalid Date

Содержание

Добро пожаловать!	1
О чём эта книга?	1
Навигация по главам	2
Используемые технологии	2
Контакты	2
Введение	2
Количественные финансы	2
О преподавателе	2
Цели курса	2
Организация курса	3
Структура курса	3
Зачем изучать финансы?	3
Почему R и Python?	3
Excel, R и Python — разные задачи	5
R или Python?	5
А что с другими инструментами?	5
Финансовые данные и их обработка	5
Получение финансовых данных	5
Yahoo Finance	6
Получение данных из Yahoo Finance с помощью <code>yfinance</code>	6
Установка и подключение	6
Задание: анализ финансовых показателей	6
Библиография	7

Добро пожаловать!

Эта книга — практический курс по количественным финансам с примерами на **R** и **Python**. Она предназначена для студентов, аналитиков и всех, кто хочет научиться применять количественные методы в финансовом анализе.

О чём эта книга?





- Как обрабатывать и визуализировать финансовые данные
- Как строить статистические и эконометрические модели
- Как оценивать рыночный риск (VaR, ES, GARCH)
- Как работать с финансовыми временными рядами

- Как анализировать акции и облигации
 - Как использовать R и Python на практике
-

Навигация по главам

- Введение
 - ...
-

Используемые технологии

-  R / Python
 -  Quarto для публикации
 -  GitHub Pages для хостинга
 -  Воспроизводимые исследования (Reproducible Research)
-

Контакты

Марсель Салихов

Директор Института энергетики и финансов, преподаватель ВШЭ

Книга распространяется под лицензией CC BY-SA 4.0


Введение

Количественные финансы

Курс «Количественные финансы» направлен на развитие практических навыков анализа данных и построения моделей с использованием реальных финансовых данных. Он рассчитан на студентов, стремящихся не только понять теоретические основы, но и научиться применять количественные методы в реальных задачах: от оценки рыночного риска до анализа финансовой отчетности и построения инвестиционных стратегий.

О преподавателе

Автор курса — **Марсель Салихов**, директор Института энергетики и финансов, преподаватель ВШЭ с 2008 года. Более 20 лет опыта в аналитике, консалтинге и прикладных исследованиях. Работает на стыке макроэкономики, финансовых рынков и энергетической политики. Пишет на R с 2012 года и на Python с 2017 года.

 Профиль на сайте ВШЭ

 Сайт Института энергетики и финансов

Цели курса

- Научиться обрабатывать и визуализировать финансовые данные.
- Освоить эконометрические методы в прикладном контексте.
- Понимать преимущества и ограничения инструментов анализа: Excel, R, Python.
- Работать с российскими и международными источниками данных.

- Приобрести навыки построения моделей, пригодных для автоматизации и воспроизводимых расчетов.

Организация курса

- **Формат:** лекции + практические занятия (лабораторные работы на Python или R).
- **Инструменты:** Jupyter/Colab, RStudio, GitHub.
- **Фокус:** применение на практике, особенно к российским данным.
- **Оценка:** лабораторные (50%), письменный экзамен (40%), посещаемость (10%).

Структура курса

1. Повторение основ Python/R: переменные, функции, структуры данных
2. Загрузка и визуализация финансовых данных
3. Обработка бухгалтерской отчетности и извлечение метрик
4. Линейные модели временных рядов (AR, MA, ARMA)
5. CAPM и бета-оценка
6. Риск-менеджмент: VaR, Expected Shortfall, GARCH
7. Портфельная теория: оптимизация, границы эффективности
8. Анализ и оценка облигаций, спот-ставки, доходности

Зачем изучать финансы?

Финансовая грамотность важна не только для профессионалов, но и для всех, кто принимает решения о сбережениях, инвестициях и страховании. Финансовая система — неотъемлемая часть современной экономики, а умение интерпретировать и анализировать данные даёт конкурентное преимущество в любой отрасли.

Почему R и Python?

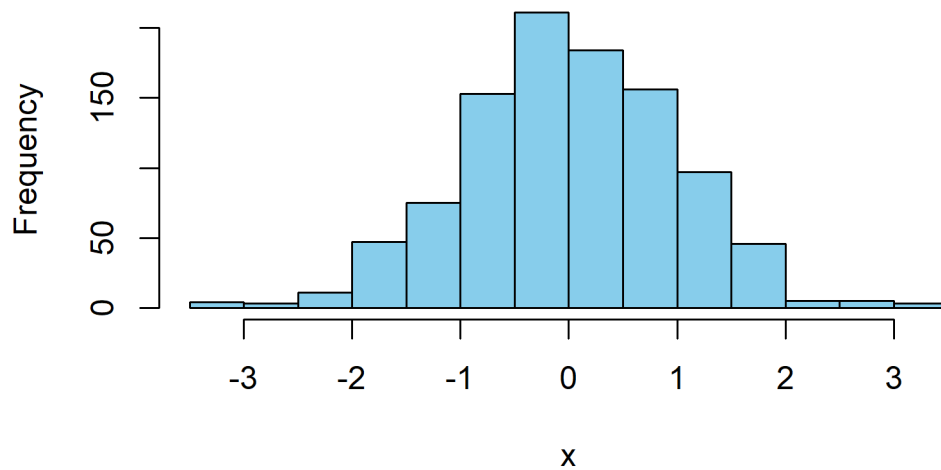
- Устоявшиеся, простые и гибкие языки — де-факто стандарт в Data Science.
- Возможность автоматизации сложных задач и повышение воспроизводимости результатов.
- Огромная экосистема пакетов: от статистики до машинного обучения.
- Поддержка интерактивной визуализации и публикации результатов.
- Доступ к исходному коду — понимание работы методов «под капотом».

Пример R-кода

```
x <- rnorm(1000)
```

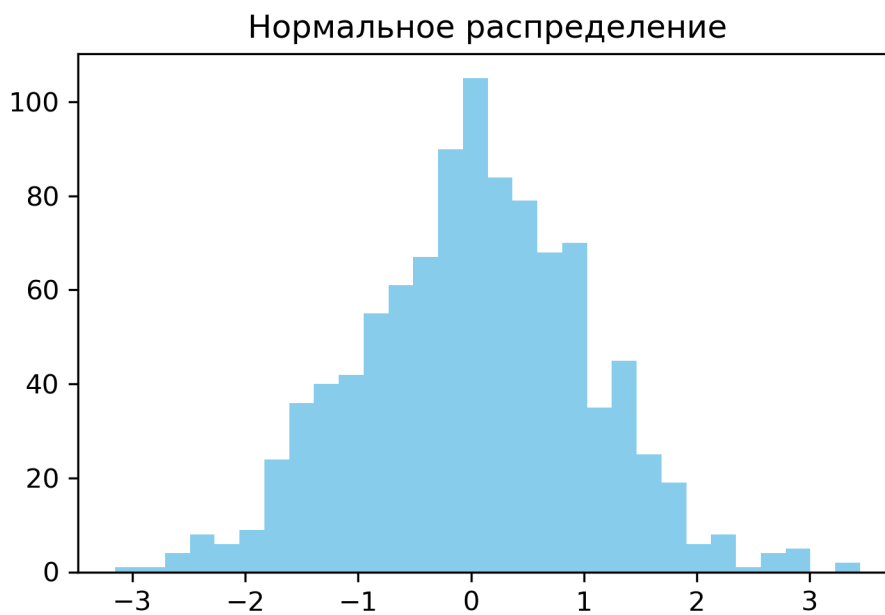
```
hist(x, col = "skyblue", main = "Нормальное распределение")
```

Нормальное распределение



```
# Пример Python-кода
import numpy as np
import matplotlib.pyplot as plt

x = np.random.normal(size=1000)
plt.hist(x, bins=30, color="skyblue")
plt.title("Нормальное распределение")
plt.show()
```



Excel, R и Python — разные задачи

Excel остаётся важным инструментом для оперативного анализа и построения простых моделей. Однако:

- Excel ограничен по объёму данных и возможностям автоматизации.
- В нём трудно находить ошибки в сложных расчетах.
- Статистические возможности Excel существенно уступают R/Python.

R и Python лучше подходят для:

- Обработки больших объемов данных
- Построения статистических и эконометрических моделей
- Визуализации и отчетности
- Воспроизводимых исследований (reproducible research)

R или Python?

Критерий	Python	R
Универсальность	+++ (общий язык программирования)	++ (ориентирован на статистику)
Гибкость/оптимизация	++	++
Визуализация	++ (plotly, matplotlib, seaborn)	+++ (ggplot2, plotly)
Популярность	+++ (1 место в TIOBE Index)	++ (18 место)

Python чаще выбирают как основной язык, особенно для анализа и машинного обучения. R — более удобен для глубокого статистического анализа, особенно в академической среде.

А что с другими инструментами?

Matlab, Stata, SPSS, EViews — важные, но в основном коммерческие инструменты. Их использование ограничено лицензиями, меньшей гибкостью и, как правило, не открытым исходным кодом. По сравнению с ними, R и Python предлагают более широкий и доступный набор возможностей.

В следующих главах мы перейдём к практическому анализу данных, начав с загрузки, визуализации и базовой статистики финансовых временных рядов.

Финансовые данные и их обработка

Получение финансовых данных

Любой количественный анализ начинается с данных. От их качества напрямую зависит достоверность результатов. Недаром в анализе данных существует выражение: «Что вложишь, то и получишь» (Garbage in — garbage out).

Перед тем как строить модели, рассчитывать риски или оптимизировать портфели, необходимо загрузить финансовые данные из надежных источников и привести их в пригодный для анализа формат.

Yahoo Finance

Вот черновик раздела на русском языке для вашей книги «Количественные финансы» — глава «Получение данных из Yahoo Finance»:

Получение данных из Yahoo Finance с помощью yfinance

Для количественного анализа в финансах критически важно иметь доступ к историческим и фундаментальным данным компаний. Один из самых удобных способов получить такие данные — использовать библиотеку `yfinance`, которая предоставляет доступ к данным с Yahoo Finance.

Yahoo Finance содержит широкий спектр информации: котировки акций, дивиденды, финансовую отчетность компаний (отчеты о прибылях и убытках, балансы, отчеты о движении денежных средств), данные по опционам и рекомендации аналитиков.

Установка и подключение

```
import yfinance as yf
```

Для начала работы достаточно создать объект `Ticker`, указав тикер интересующей компании:

```
AAPL = yf.Ticker("AAPL")
```

Теперь у нас есть доступ ко всей информации, связанной с Apple Inc.

Например, чтобы получить отчет о прибылях и убытках:

```
AAPL.financials
```

Список всех доступных атрибутов можно посмотреть через:

```
dir(AAPL)
```

Задание: анализ финансовых показателей

Давайте решим практическую задачу на основе полученных данных:

Цель:

1. Найти значения выручки (`revenue`), чистой прибыли (`net income`) и операционных расходов (`operating expenses`) за последние годы.
2. Представить данные в виде удобной таблицы.
3. Оценить динамику чистой прибыли по годам и рассчитать её годовые темпы роста (или падения).
4. Вычислить отношение долга к капиталу (`debt-to-equity ratio`), если оно доступно в отчетности.

Пример кода:

```
import pandas as pd
```

```
# Получение финансовой отчетности
```

```

income_statement = AAPL.financials
balance_sheet = AAPL.balance_sheet

# Транспонируем для удобства чтения
df = income_statement.T

# Отобираем нужные строки
df_selected = df[["Total Revenue", "Net Income", "Operating Expenses"]]

# Добавим расчет изменения чистой прибыли
df_selected["Net Income YoY %"] = df_selected["Net Income"].pct_change() * 100

# Добавим расчет коэффициента debt-to-equity, если доступны данные
debt = balance_sheet.loc["Total Liab"]
equity = balance_sheet.loc["Total Stockholder Equity"]
df_selected["Debt/Equity"] = (debt / equity).values

# Округлим и отобразим
df_selected = df_selected.round(2)
df_selected

```

Интерпретация:

- **Выручка** отражает общий объем продаж.
- **Операционные расходы** показывают, сколько компания тратит на свою деятельность.
- **Чистая прибыль** — итоговая прибыль после всех расходов и налогов.
- **Debt/Equity** — ключевой показатель финансовой устойчивости: чем выше, тем больше компания зависит от заемного капитала.

Библиография