

## ГЛАВА 1

---

# Финансы и Python

История теории финансов — интересный пример взаимосвязи между абстрактным теоретизированием и практическим применением.

*Франк Мильне (1995)*

Благодаря современным технологиям в последние годы хедж-фонды поглотили десятки миллиардов долларов инвестиций. Эти же технологии приносят прибыль людям, принимающим финансовые решения в таких организациях.

*Лоуренс Флетчер (2020)*

Цель главы 1 состоит в том, чтобы заложить основы для последующих глав, поэтому в ней дается краткий обзор освещаемых тем. Она начинается с раздела «Краткая история финансов», в котором в общих чертах рассказывается об истории развития и современном состоянии финансов. В разделе «Главные тенденции в области финансов» рассматриваются основные тенденции, касающиеся математики, технологий, данных, искусственного интеллекта и определяющие развитие финансов в долгосрочной перспективе. В связи с этим раздел «Четырехязычная сфера» говорит о том, что финансы в наше время — это дисциплина, состоящая из четырех тесно связанных между собой типов языков: естественного (в большей мере английского), языка финансов, математического и языка программирования. Далее объясняется, как устроена книга (раздел «Структура книги»). А в последнем разделе, «Вводная информа-

ция о Python», рассказывается об установке окружения Python на компьютер и об альтернативном варианте работы с ним: весь код может использоваться и исполняться через обычный браузер на Quant Platform.

## Краткая история финансов

Чтобы лучше понять текущее состояние финансов и финансовой отрасли, нужно взглянуть, как они развивались. В 2006 году Рубинштейн условно разделил историю финансов как науки на три периода.

- *Древний период (до 1950 года)*. Период, характеризующийся в основном неформальными рассуждениями, эмпирическими правилами и опытом субъектов финансового рынка.
- *Классический период (1950–1980 годы)*. В это время произошло внедрение в финансы более строгих обоснований и математики. Были разработаны финансовые модели, например модель ценообразования опционов Блэка — Шоулза (1973), а также заложены некоторые основы, например риск-нейтральный подход к ценообразованию Харрисона и Крепса (1979).
- *Современный период (1980–2000 годы)*. Это период прогресса в отдельных областях финансов, например в финансовой инженерии, и решения проблем, связанных с важными эмпирическими явлениями на финансовых рынках, такими как стохастическая модель процентных ставок (Кокс, Ингерсолл и Росс, 1985) или модель стохастической волатильности (Хестон, 1993).

Сейчас, спустя 15 лет после публикации книги Рубинштейна, можно выделить четвертый и пятый периоды, которые обеспечили появление и нынешнюю популярность Python в финансах.

- *Вычислительный период (2000–2020 годы)*. В это время произошел переход от теоретической направленности финансов к вычислительной, что обусловлено развитием программно-технических средств. Эту смену парадигмы хорошо иллюстрирует численный алгоритм для оценки американских опционов методом Монте-Карло, представленный в работе Лонгстаффа и Шварца в 2001 году. Алгоритм требует больших вычислительных ресурсов, поскольку для оценки стоимости всего одного опциона нужны сотни тысяч численных моделирований и анализ множества обычных регрессий методом наименьших квадратов (см.: Хилпиш, 2018).

- *Период использования искусственного интеллекта (после 2020 года).* Прогресс в области искусственного интеллекта (ИИ) и связанные с ним истории успеха подстегнули интерес к использованию возможностей ИИ в финансовой сфере. Несмотря на существование эффективных приложений в данной области (см.: Хилпиш, 2020), можно предположить, что с 2020 года парадигма планомерно смещается в сторону *финансовых систем, основанных на ИИ*. Таким образом, мы становимся свидетелями перехода от простых, в основном линейных, финансовых моделей к сложным моделям и использованию алгоритмов ИИ (глубоких нейронных сетей или обучения с подкреплением) для фиксации, описания и объяснения финансовых явлений.

## Главные тенденции в области финансов

Как и многие другие предметы и отрасли, наука о финансах со временем стала формализованной научной дисциплиной. Этому прогрессу поспособствовали растущая популярность формальной математики, развитие технологий, доступность данных и усовершенствованные алгоритмы, например ИИ-алгоритмы. В целом в эволюции финансов можно выделить четыре основных направления.

- *Математика.* Начиная с 1950-х годов наука о финансах становится все более формализованной дисциплиной, в которой систематически используются знания из различных областей математики, например линейной алгебры и стохастического исчисления. Появление портфельной теории Марковица (1952) стало прорывом в количественном анализе и точкой перехода от древнего периода с его неформальными рассуждениями к классическому периоду.
- *Технологии.* Благодаря рабочим станциям, серверам и компьютерам, которые начали широко использоваться в конце 1980-х — начале 1990-х годов, технологии стали проникать и в финансовую отрасль. И если сначала техника имела довольно небольшую вычислительную мощность, то сейчас даже самые сложные финансовые задачи можно решить с помощью специальной программы, не прибегая к специальным моделям и методам, характерным для классического и современного периодов. Кредо стало таким: «Обновляйте оборудование и используйте современное программное обеспечение вместе с правильными вычислительными методами». При этом большинство современных компьютеров, предназначенных для

использования массовым потребителем, уже обладает мощностью, необходимой для высокопроизводительной работы (например, для параллельной обработки), что значительно упрощает изучение финансовой инженерии и взаимодействие с финансовыми системами, основанными на ИИ.

- *Данные.* Если в древний и классический периоды теории и практики черпали информацию, касающуюся финансов, в основном из специализированных печатных изданий (здесь можно вспомнить Wall Street Journal или Financial Times), то сегодня возросла доступность массива финансовых данных в электронном виде. Массивы высокочастотных внутридневных данных стали нормой и заменили цены на момент закрытия биржи в качестве основной базы для эмпирических исследований. Каждый торговый день одна акция может генерировать массивы внутридневных данных, содержащих более 100 000 значений, что приблизительно эквивалентно ценам на момент закрытия биржи для той же акции за 400 лет (250 торговых дней в году умножить на 400 лет). Вдобавок с недавнего времени наблюдается распространение открытых массивов данных, что также значительно снижает порог входа в финансовую инженерию, алгоритмическую торговлю или финансовую эконометрику.
- *Искусственный интеллект.* Увеличение объема финансовых данных (возникновение «больших финансовых данных») в наши дни делает применение алгоритмов искусственного интеллекта: алгоритмов машинного обучения, глубокого обучения или обучения с подкреплением (см.: Хилпиш, 2020) — не только возможным, но и во многих случаях необходимым подходом к работе с финансами. Традиционные статистические методы из финансовой эконометрики уже не подходят для решения большинства современных проблем, возникающих на финансовых рынках. В условиях нелинейной, многомерной, постоянно меняющейся финансовой среды алгоритмы на основе искусственного интеллекта нередко становятся единственным средством обнаружения релевантных взаимосвязей и закономерностей, получения важных данных и использования улучшенных возможностей прогнозирования.

В книге дается основная информация о финансовой математике и современных технологиях, используемых для реализации формальных финансовых моделей. С ее помощью можно приобрести навыки работы с массивами финансовых данных, наиболее часто встречающимися в финансовой сфере. Книга предназначена для того, чтобы подготовить читателя к более сложным темам финансовой инженерии и применения ИИ к финансовым расчетам.



### Python и финансы

Все чаще основой финансовых расчетов становятся алгоритмы, требующие больших вычислительных ресурсов, постоянно растущая доступность данных и искусственный интеллект. Python оказался подходящим языком программирования и технологической платформой для удовлетворения потребностей и решения задач, обусловленных тенденциями развития финансовой сферы.

## Четырехязычная сфера

На фоне перечисленных выше тенденций финансы стали сферой применения четырех языков.

- *Естественный язык.* Сегодня основным языком публикуемых финансовых исследований, книг, статей и новостей является английский.
- *Финансовый язык.* Как и в любой другой области, в финансах есть свои технические термины, понятия и выражения, описывающие определенные финансовые явления или понятия.
- *Математический язык.* Математика — самый удобный языковой инструмент для формализации финансовых понятий и концепций.
- *Язык программирования.* Как отмечается в эпиграфе к введению, Python — лучший из существующих языков программирования для работы в финансовой индустрии.

Таким образом, чтобы отлично разбираться в науке о финансах, и теоретик, и практик должны свободно владеть всеми четырьмя языками. Конечно, нельзя утверждать, что в финансовой области нет других языков, кроме английского или, например, Python. Здесь подразумевается, что при ограниченном количестве времени, выделенном на изучение финансов, наряду с финансовой математикой лучше сосредоточить свои силы на Python, а не на другом языке программирования.

## Структура книги

Как в издании совмещается использование всех этих четырех языков? С естественным языком все понятно: можно либо читать книгу в оригинале, либо пользоваться переводом на родной язык. Осталось решить вопрос с остальными.

К примеру, книга не может подробно объяснить каждый отдельный раздел математики, необходимый в финансах, или представить все концепции программирования (Python), использующиеся в финансовой инженерии. Тем не менее в ней была сделана попытка связать понятия из финансов, математики и программирования между собой.

Начиная с главы 2, информация в книге будет подаваться через введение финансового понятия (или концепции) вместе с его математическим представлением и реализацией на Python. К примеру, в таблице из главы 3 перечислены финансовые понятия, о которых будет идти речь, соответствующие им основные математические элементы и основная структура данных Python, используемая для реализации финансовой математики.

Финансы	Математика	Python
Неопределенность	Вероятностное пространство	ndarray
Финансовые активы	Векторы, матрицы	ndarray
Достижимые условные требования	Линейная оболочка векторов, базис векторного пространства	ndarray

Чтобы проиллюстрировать общую структуру книги, далее приведу пример описания финансового понятия, который будет рассматриваться подробнее в следующих главах.

Возьмем из таблицы концепцию *неопределенности в финансах*. Под неопределенностью понимается невозможность заранее узнать состояние модельной экономики в будущем. Будущее состояние экономики важно, например, для определения выплаты по европейскому колл-опциону (то есть по опциону на покупку). В условиях дискретности мы имеем дело с конечным числом таких состояний, например двумя, тремя или более. В самом простом случае, когда рассматриваются только два будущих состояния, выплата по европейскому колл-опциону математически представляется как *случайная величина*, которая, в свою очередь, формально может быть представлена вектором  $\mathbf{v}$ , являющимся элементом *векторного пространства*  $\mathbb{R}^2$ . Векторное пространство — это набор объектов (векторов), для которых определены сложение и скалярное умножение. Формально вектор  $\mathbf{v}$  можно представить следующим образом:

$$\mathbf{v} = \begin{pmatrix} v^u \\ v^d \end{pmatrix} \in \mathbb{R}_{\geq 0}^2$$

Здесь предполагается, что оба векторных элемента являются неотрицательными вещественными (действительными) числами  $v^u, v^d \in \mathbb{R}_{\geq 0}$ . Другими словами, если неопределенная, зависящая от состояния экономики цена акций, на которые выписан европейский колл-опцион, определяется как:

$$S = \begin{pmatrix} 20 \\ 5 \end{pmatrix} \in \mathbb{R}_{\geq 0}^2,$$

а цена исполнения опциона  $K = 15$ , то выплата  $C$  по европейскому колл-опциону будет равна:

$$C = \max(S - K, 0) = \begin{pmatrix} \max(20 - 15, 0) \\ \max(5 - 15, 0) \end{pmatrix} = \begin{pmatrix} 5 \\ 0 \end{pmatrix} \in \mathbb{R}_{\geq 0}^2.$$

Вот так понятия *неопределенной цены акции* и *зависящей от состояния экономики выплаты по европейскому опциону* могут быть математически смоделированы в виде вектора. Дисциплина, занимающаяся векторами и векторными пространствами в математике, называется *линейной алгеброй*.

Как же все это перевести на Python? Во-первых, вещественные числа в нем представлены как *числа с плавающей запятой* или объекты `float`:

```
In [1]: vu = 1.5 ❶
```

```
In [2]: vd = 3.75 ❷
```

```
In [3]: type(vu) ❸
```

```
Out[3]: float
```

```
In [4]: vu + vd ❹
```

```
Out[4]: 5.25
```

- ❶ Определение переменной `vu` и ее значения 1.5.
- ❷ Определение переменной `vd` и ее значения 3.75.
- ❸ Поиск и вывод типа объекта `vu` (`float`).
- ❹ Сложение значений `vu` и `vd`.

Во-вторых, наборы объектов одного типа в программировании обычно называются *массивами*. Поддержку таких структур данных в Python обеспечивает библиотека NumPy (<http://numpy.org/>). Основной структурой данных в ней

является `ndarray` — аббревиатура для *n*-мерного массива (*n*-dimensional array). NumPy легко моделирует векторы с вещественными значениями:

```
In [5]: import numpy as np ❶
```

```
In [6]: v = np.array((vu, vd)) ❷
```

```
In [7]: v ❸  
Out[7]: array([1.5 , 3.75])
```

```
In [8]: v.dtype ❹  
Out[8]: dtype('float64')
```

```
In [9]: v.shape ❺  
Out[9]: (2,)
```

```
In [10]: v + v ❻  
Out[10]: array([3. , 7.5])
```

```
In [11]: 3 * v ❼  
Out[11]: array([ 4.5 , 11.25])
```

- ❶ Импорт библиотеки NumPy.
- ❷ Создание объекта `ndarray`.
- ❸ Вывод хранимых в объекте данных.
- ❹ Поиск и вывод типа данных всех элементов.
- ❺ Поиск и вывод формы объекта.
- ❻ Сложение векторов.
- ❼ Скалярное умножение.

Мы увидели работу Python с математическими понятиями, связанными с векторами. Осталось рассмотреть применение этих возможностей к финансам:

```
In [12]: S = np.array((20, 5)) ❶
```

```
In [13]: K = 15 ❷
```

```
In [14]: C = np.maximum(S - K, 0) ❸
```

```
In [15]: C ❹  
Out[15]: array([5, 0])
```



- ❶ Обозначение неопределенной цены акции как объект `ndarray`.
- ❷ Определение цены исполнения опциона через переменную Python с целочисленным значением (объект `int`).
- ❸ Поэлементное вычисление функции максимума.
- ❹ Итоговые данные, хранящиеся в объекте `ndarray`, обозначенном `C`.

Таким образом, при написании книги использовался следующий подход.

1. Вводятся финансовые понятия и концепции.
2. Дается их математическое представление и модель.
3. Математическая модель переводится в исполняемый код Python.

В этом плане наука о финансах обосновывает использование математики, которая, в свою очередь, объясняет применение методов программирования на Python.

## Вводная информация о Python

Одним из преимуществ Python является открытый исходный код, который облегчает его установку на все основные операционные системы — macOS, Windows и Linux. При этом для работы с кодом из этой книги и финансов в целом в дополнение к базовому интерпретатору Python необходимы всего несколько основных пакетов и библиотек, тоже имеющих открытый исходный код.

- NumPy (<http://numpy.org/>). Позволяет эффективно работать с большими  $n$ -мерными массивами числовых данных.
- pandas (<http://pandas.pydata.org/>). Предназначен в первую очередь для эффективной работы с табличными наборами данных и финансовыми временными рядами. В данной книге pandas не будет задействован, однако стоит отметить его особую популярность в области финансов.
- SciPy (<http://scipy.org/>). Является набором научных функций, необходимых, например, для решения типичных задач, связанных с оптимизацией.

- `SymPy` (<http://sympy.org/>). Позволяет использовать символьную математику, что иногда бывает полезно в работе с финансовыми моделями и алгоритмами.
- `matplotlib` (<http://matplotlib.org/>). Представляет собой стандартную библиотеку Python для визуализации данных. Она позволяет создавать и настраивать различные типы графиков, например линейные графики, столбчатые диаграммы и гистограммы.

Кроме того, для начала работы с интерактивным кодированием на Python требуются еще два инструмента.

- *IPython* (<http://ipython.org/>). Самая популярная среда для интерактивного кодирования на Python в командной строке (в терминале, оболочке `shell`).
- *JupyterLab* (<http://jupyter.org/>). Интерактивная среда для интерактивного кодирования и разработки на Python в браузере.

Технические требования для изучения программирования на Python минимальны. Есть два варианта запуска кода Python.

- *Quant Platform*. На бесплатной платформе *Quant Platform* (<http://finpy.pqp.io/>) находится полноценная среда для интерактивной финансовой аналитики с помощью Python. На ней можно запускать код из данной книги через браузер, что делает ненужной установку программы на компьютер. После бесплатной регистрации вы получаете автоматический доступ ко всему коду и всем блокнотам *Jupyter Notebook*, используемым в книге, что дает возможность сразу выполнять код в браузере.
- *Локальная среда Python*. В настоящее время несложно установить локальную среду Python, которая позволит вам погрузиться в финансовую аналитику и выполнить код из книги на своем компьютере. Далее описано, как произвести установку.



### Выбор между локальной средой Python и *Quant Platform*

Практика показывает, что локальная установка правильной среды Python иногда может вызвать сложности у новичков в программировании. Поэтому, если на начальном этапе установка Python на компьютер вызывает какие-то проблемы, лучше не тратить слишком много времени на нее, а воспользоваться платформой *Quant Platform* (<http://finpy.pqp.io/>). К установке Python на компьютер можно вернуться позже, когда у вас будет больше опыта.