

ОШ МАМЛЕКЕТТИК УНИВЕРСИТЕТИНИН ЖАРЧЫСЫ

ВЕСТНИК ОШКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА

BULLETIN OF OSH STATE UNIVERSITY

ISSN: 1694-7452 e-ISSN: 1694-8610

№2/2025, 122-132

ИНФОРМАТИКА

УДК: 330.1:004.021

DOI: [10.52754/16948610_2025_2_11](https://doi.org/10.52754/16948610_2025_2_11)

**ТРАНСФОРМАЦИЯ ФИНАНСОВОГО АНАЛИЗА С ПОМОЩЬЮ ЯЗЫКА
ПРОГРАММИРОВАНИЯ PYTHON: ОТ РУТИННЫХ ВЫЧИСЛЕНИЙ К
СТРАТЕГИЧЕСКИМ РЕШЕНИЯМ**

PYTHON ПРОГРАММАЛОО ТИЛИН КОЛДОНУУ МЕНЕН ФИНАНСЫЛЫК АНАЛИЗДИ
ТРАНСФОРМАЦИЯЛОО: КОЛ ЭСЕПТӨӨЛӨРҮНӨН СТРАТЕГИЯЛЫК ЧЕЧИМДЕРГЕ

TRANSFORMATION OF FINANCIAL ANALYSIS USING PYTHON PROGRAMMING
LANGUAGE: FROM ROUTINE CALCULATIONS TO STRATEGIC DECISIONS

Аркабаев Нуркасым Кылычбекович

Аркабаев Нуркасым Кылычбекович

Arkabayev Nurkasym Kilychbekovich

к.ф.-м. н. доцент, Ошский государственный университет

ф.-м.и.к., доцент, Ош мамлекеттик университети

Candidate of Physico-Mathematical Sciences, Associate Professor, Osh State University

narkabaev@oshsu.kg

ORCID: 0009-0000-1912-2225

Арапбаев Байэл Мирбекович

Арапбаев Байэл Мирбекович

Arapbaev Bayel Mirbekovich

магистрант, Ошский государственный университет

магистрант, Ош мамлекеттик университети

Master's Student, Osh State University

bekeshowb@gmail.com

ORCID: 0009-0003-1844-5817

Ражапов Сыймык Каныбекович

Ражапов Сыймык Каныбекович

Razhapov Syimyk Kanybekovich

магистрант, Ошский государственный университет

магистрант, Ош мамлекеттик университети

Master's Student, Osh State University

rajapovv0105@gmail.com

ORCID: 0009-0005-9065-3738

ТРАНСФОРМАЦИЯ ФИНАНСОВОГО АНАЛИЗА С ПОМОЩЬЮ ЯЗЫКА ПРОГРАММИРОВАНИЯ PYTHON: ОТ РУТИННЫХ ВЫЧИСЛЕНИЙ К СТРАТЕГИЧЕСКИМ РЕШЕНИЯМ

Аннотация

Цифровизация экономических процессов кардинально меняет парадигму финансового анализа, трансформируя его из набора рутинных вычислений в инструмент стратегического планирования и принятия решений. Современные экономические задачи требуют оперативной обработки больших объёмов данных, что делает автоматизацию финансового анализа одной из ключевых областей применения информационных технологий. В статье рассматривается использование языка программирования Python для упрощения и ускорения финансовых расчётов, анализа данных и визуализации результатов. Основное внимание уделяется практическим примерам автоматизации таких задач, как расчёт бюджета, анализ инвестиционных портфелей и прогнозирование финансовых показателей. Применение библиотек pandas, numpy и matplotlib позволяет не только сократить время выполнения задач, но и повысить точность анализа. Авторы приводят сравнительный анализ традиционных методов обработки финансовой информации и современных подходов с использованием программирования. Результаты демонстрируют, что автоматизация с помощью Python делает финансовый анализ доступным даже для специалистов без глубоких знаний программирования, освобождая время для стратегического планирования и открывая новые возможности для принятия обоснованных бизнес-решений.

Ключевые слова: автоматизация, финансовый анализ, Python, pandas, numpy, matplotlib, экономика, данные

**PYTHON ПРОГРАММАЛОО ТИЛИН КОЛДОНУУ
МЕНЕН ФИНАНСЫЛЫК АНАЛИЗДИ
ТРАНСФОРМАЦИЯЛОО: КОЛ
ЭСЭПТӨӨЛӨРҮНӨН СТРАТЕГИЯЛЫК
ЧЕЧИМДЕРГЕ**

**TRANSFORMATION OF FINANCIAL ANALYSIS
USING PYTHON PROGRAMMING LANGUAGE:
FROM ROUTINE CALCULATIONS TO STRATEGIC
DECISIONS**

Аннотация

Экономикалык процесстердин санариптештирилиши финансылык анализдин парадигмасын түп-тамырынан бери өзгөртүп, аны кадимки эсептөөлөр топтомунан стратегиялык пландаштыруу жана чечим кабыл алуу каражатына айландырууда. Заманбап экономикалык милдеттер маалыматтардын чоң көлөмүн тез иштетүүнү талап кылат, бул финансылык анализди автоматташтырууну маалыматтык технологияларды колдонуунун негизги чөйрөлөрүнүн бирине айландырат. Макалада финансылык эсептөөлөрдү, маалыматтарды талдоону жана натыйжаларды визуалдаштырууну жөнөкөйлөштүрүү жана тездетүү үчүн Python программалоо тилин колдонуу жагы изилденет. Негизги басым бюджетти эсептөө, инвестициялык портфелдерди талдоо жана финансылык көрсөткүчтөрдү болжолдоо сыяктуу тапшырмаларды автоматташтыруунун практикалык мисалдарына коюлат. Pandas, numpy жана matplotlib сыяктуу библиотекаларды колдонуу менен маселелерди аткаруу үчүн керектүү убакытты кыскартып гана тим болбостон, анализдин тактыгын да жогорулатат. Авторлор финансылык маалыматтарды салттуу иштетүү ыкмаларынын жана программалоону колдонуу менен заманбап ыкмалардын салыштырма анализин мисал катары карашат. Жыйынтыгында Python менен автоматташтыруу финансылык анализди программалоо боюнча билими жок адистер үчүн да жеткиликтүү кылаарын, стратегиялык пландаштыруу үчүн убакытты үнөмдөп, ишкердик чечимдерди кабыл алуу үчүн жаңы мүмкүнчүлүктөрдү жаратаарын көрсөтүшөт.

Ачкыч сөздөр: автоматташтыруу, финансылык анализ, Python, pandas, numpy, matplotlib, экономика, берилгендер

Abstract

The digitalization of economic processes is radically changing the paradigm of financial analysis, transforming it from a set of routine calculations into a tool for strategic planning and decision-making. Modern economic challenges require the rapid processing of large volumes of data, making the automation of financial analysis one of the key areas for the application of information technologies. This article explores the use of the Python programming language to simplify and accelerate financial calculations, data analysis, and result visualization. The focus is placed on practical examples of automating tasks such as budgeting, investment portfolio analysis, and financial forecasting. The use of libraries such as pandas, numpy, and matplotlib not only reduces the time needed to complete tasks but also improves the accuracy of the analysis. The authors provide a comparative analysis of traditional financial data processing methods versus modern programming approaches. The results show that automation with Python makes financial analysis accessible even to professionals without deep programming knowledge, freeing up time for strategic planning and opening new opportunities for informed business decisions.

Keywords: Automation, financial analysis, Python, pandas, numpy, matplotlib, economics, data

Введение

Финансовый анализ – это сердце принятия решений в бизнесе, инвестициях и управлении ресурсами. Однако традиционные методы, такие как работа с таблицами в Excel или ручные расчёты, становятся всё менее эффективными в условиях роста объёмов данных. Представьте себе аналитика, который тратит часы на сведение баланса или прогнозирование доходов, когда эти задачи можно решить за минуты. Именно здесь на сцену выходит Python – язык программирования, который сочетает простоту освоения с мощными инструментами для обработки данных (Наралиев, Имаралиев и Бакытбек кызы, 2021).

Как отмечают Борисова и Дашин (2025, 168) в своём исследовании, модернизация систем анализа финансовых показателей предоставляет руководству возможность эффективно обрабатывать большие объёмы информации, оперативно выявлять проблемные аспекты в деятельности организации и принимать своевременные меры по их устранению, а также снижать негативное влияние меняющихся внешних и внутренних факторов на результативность предприятия, что в конечном итоге способствует укреплению конкурентных позиций и расширению рыночной доли в соответствующем сегменте. Суханова (2021, 38) в своём исследовании пишет о том что, автоматизация анализа финансового состояния компании не только исключает ошибки, допущенные по причине человеческого фактора, но и позволяет сократить время обработки информации, снизить трудозатраты и высвободить финансовые ресурсы за счёт исключения участия специалистов различных областей в процессе анализа, что особенно актуально для компаний электросетевого комплекса с их разветвлённой структурой финансовых показателей.

Python уже давно перестал быть инструментом исключительно для программистов. Сегодня это универсальный помощник для экономистов, аналитиков и финансистов, которые хотят ускорить свою работу и сосредоточиться на стратегических задачах, а не на рутине. В этой статье мы рассмотрим, как Python помогает автоматизировать финансовый анализ, делая его быстрее, точнее и доступнее. Как отмечает VanderPlas (2016, 3), именно комбинация доступности языка и мощного экосистемного окружения делает Python оптимальным выбором для финансовых специалистов, не имеющих глубокого технического бэкграунда. По мнению Шаре (1994, 54), автоматизация расчета финансовых коэффициентов не только экономит время, но и обеспечивает методологическую строгость, недостижимую при использовании электронных таблиц.

Мы рассмотрим несколько практических примеров, которые показывают, как с помощью библиотек *pandas*, *numpy* и *matplotlib* можно решать реальные задачи, такие как управление бюджетом, анализ портфелей и прогнозирование. Наша цель – показать, что автоматизация – это не только про технологии, но и про освобождение времени для более важных идей. Челухина и Асяева (2021, 1041) в своем исследовании приходят к важному выводу: финансовые специалисты, внедрившие Python-автоматизацию в свою работу, тратят до 40% больше времени на стратегическое планирование и качественный анализ, что напрямую влияет на эффективность бизнес-решений. Анализируя тенденции автоматизации оптимизация складского учета с использованием технологий интернета вещей и платформы .NET авторов Аркабаева и др. (2024, 153) обсуждаются практические результаты внедрения, включая автоматизацию инвентаризации, оптимизацию размещения товаров и улучшенное управление цепочками поставок, и оптимизация средствами платформы .NET.

Рассматриваются вызовы, связанные с безопасностью данных и интеграцией с существующими системами.

Как подчеркивают Имаралиев и соавторы (2025, 159), математические методы и компьютерное моделирование являются универсальными инструментами для решения задач оптимизации экономических процессов, что особенно актуально при трансформации финансового анализа с использованием современных языков программирования.

Важно отметить, что интеграция Python в экономические и финансовые процессы происходит на фоне общей цифровизации финансовой сферы. Финансовые технологии (FinTech) активно развиваются, и Python становится их неотъемлемой частью благодаря своей гибкости и обширной экосистеме библиотек. Особенно это актуально в условиях растущей конкуренции, когда скорость принятия решений часто определяет успех организации. Финансовые аналитики, освоившие Python, получают значительное преимущество в виде возможности быстрее и глубже анализировать данные, что в свою очередь ведет к более обоснованным финансовым решениям. Как подчеркивают Аарон (Электронный ресурс, 2025, Режим доступа: <https://ru.bitdegree.org/rukovodstvo/python-dlya-finansov>) и Фолкман (Электронный ресурс, 2025, Режим доступа: <https://towardsdatascience.com/stop-using-excel-for-data-analytics-upgrade-to-python-46b5963fb036/>) в своих работах, даже базовые знания Python и его специализированных библиотек позволяют финансовым специалистам автоматизировать до 70% рутинных операций, что не только сокращает время на подготовку отчетности, но и значительно снижает вероятность ошибок, неизбежно возникающих при ручной обработке больших массивов данных.

В работе Аркабаева и др. (2024, 295) рассматривается оптимизация логистических процессов и решение транспортных задач в контексте изменяющихся потребностей современного рынка с применением библиотеки Java и представлены ключевые инновационные подходы, такие как использование технологий интернета вещей, искусственного интеллекта и аналитики больших данных для улучшения управления маршрутами, уменьшения излишних затрат. Следует отметить, что при автоматизации финансового анализа особую перспективность демонстрирует применение нейронных сетей. Глубокое обучение позволяет выявлять сложные нелинейные зависимости в финансовых данных, недоступные для обнаружения традиционными статистическими методами. Библиотеки Python, такие как TensorFlow и PyTorch, предоставляют финансовым аналитикам доступные инструменты для разработки моделей, способных прогнозировать рыночные тренды, оценивать кредитные риски и выявлять аномалии в транзакционных потоках. Особенно эффективными оказываются рекуррентные нейронные сети (RNN) и их модификации (LSTM, GRU) при анализе временных рядов финансовых показателей, что открывает новые возможности для точного прогнозирования поведения рынков и оптимизации инвестиционных стратегий.

Постановка задачи

Автоматизация финансового анализа начинается с понимания, что именно мы хотим упростить. В реальной жизни аналитик сталкивается с множеством задач: от расчёта доходности инвестиций до построения прогнозов на основе исторических данных. Эти процессы часто требуют работы с большими таблицами, сложных вычислений и наглядной

визуализации, чтобы результаты были понятны не только специалисту, но и, скажем, руководителю или клиенту.

Основная цель – сократить время на выполнение рутинных операций, минимизировать ошибки и сделать анализ воспроизводимым. Например, если вы каждый месяц обновляете отчёт о расходах, то вместо того чтобы вручную копировать данные в таблицы, можно написать скрипт, который сделает это за вас. Или, если нужно оценить риски инвестиционного портфеля, Python может быстро рассчитать ключевые показатели, такие как волатильность или коэффициент Шарпа, и показать их в удобной форме.

Для решения этих задач мы выбрали три направления: расчёт бюджета, анализ инвестиционного портфеля и базовое прогнозирование. Эти примеры охватывают типичные сценарии, с которыми сталкиваются экономисты, и показывают, как Python может стать их союзником. Мы будем использовать библиотеки pandas для работы с данными, numpy для вычислений и matplotlib для визуализации, потому что они просты в освоении и невероятно мощны.

Методы

Работа с данными: pandas как основа анализа

Следует подчеркнуть, что работа с pandas в финансовом анализе выходит за рамки простых операций с данными. Одна из сильных сторон этой библиотеки – возможность легко обрабатывать временные ряды, что критически важно для финансового анализа. Функции `resample()` и `rolling()` позволяют агрегировать данные по различным временным интервалам и вычислять скользящие средние, что делает анализ трендов интуитивно понятным. Кроме того, метод `pivot_table()` трансформирует данные в формат, удобный для многомерного анализа, что особенно полезно при сравнении финансовых показателей по нескольким измерениям одновременно – например, по категориям расходов и временным периодам.

Первый шаг в любом финансовом анализе – это работа с данными. Будь то таблица расходов, список транзакций или данные о ценах акций, всё начинается с их обработки. Библиотека pandas – это как швейцарский нож для аналитика: она позволяет загружать данные из разных источников, чистить их, сортировать и трансформировать так, как нужно.

Допустим, у нас есть таблица с данными о расходах компании за год. В ней есть столбцы с категориями (например, "маркетинг", "зарплаты"), суммами и датами. Вместо того чтобы вручную фильтровать данные в Excel, мы можем загрузить таблицу в pandas и за пару строк кода получить, скажем, суммарные расходы по каждой категории или выявить месяцы с аномально высокими тратами. Pandas делает такие операции интуитивными: вы просто говорите, что хотите, а библиотека делает остальное.

Кроме того, pandas отлично справляется с объединением данных из разных источников. Например, если у вас есть данные о продажах в одном файле, а о расходах – в другом, вы можете легко соединить их по общему ключу, например, по датам. Это особенно полезно, когда нужно построить целостную картину финансового состояния.

Вычисления: numpy для скорости и точности

Когда данные подготовлены, наступает время расчётов. Здесь в игру вступает numpy – библиотека, которая делает числовые операции быстрыми и надёжными. Она особенно

полезна, когда нужно работать с большими массивами данных, например, рассчитывать доходность портфеля из сотен активов или моделировать финансовые сценарии.

Возьмём пример с инвестиционным портфелем. Чтобы оценить его эффективность, аналитик может захотеть рассчитать среднюю доходность, стандартное отклонение (как меру риска) или корреляцию между активами. NumPy позволяет сделать это буквально в несколько строк кода, причём с высокой скоростью даже на больших датасетах. Например, вычисление коэффициента Шарпа – показателя, который сравнивает доходность портфеля с его риском, – становится тривиальной задачей, если использовать numpy для расчёта среднего и стандартного отклонения

Преимущество numpy особенно ярко проявляется при работе с матричными вычислениями, которые лежат в основе многих финансовых моделей. Например, при оптимизации портфеля по методу Марковица требуется рассчитывать ковариационные матрицы и решать системы линейных уравнений – задачи, для которых numpy предлагает высокоэффективные функции. Важно также отметить, что numpy обеспечивает стабильность вычислений даже в случаях с неустойчивыми финансовыми данными, применяя методы численной стабилизации, что критически важно при работе с волатильными рынками или длинными временными рядами.

Визуализация: matplotlib для наглядности

Данные и расчёты – это только половина дела. Чтобы результаты анализа дошли до аудитории, их нужно представить в понятной форме. Здесь нам помогает matplotlib – библиотека для создания графиков и диаграмм. С её помощью можно построить всё: от простых линейных графиков, показывающих динамику доходов, до сложных тепловых карт, иллюстрирующих корреляцию активов.

Представьте, что вы хотите показать руководству, как изменились расходы компании за год. Вместо того чтобы отправлять им таблицу с цифрами, вы можете построить график, где каждая категория расходов представлена отдельной линией. Или, если речь идёт об инвестициях, круговая диаграмма покажет, как распределяются активы в портфеле. Matplotlib позволяет настроить графики так, чтобы они были не только информативными, но и красивыми, что особенно важно, когда нужно презентовать результаты.

Практические примеры

Теперь давайте посмотрим, как всё это работает на практике. Мы выбрали три сценария, которые часто встречаются в финансовом анализе, и покажу, как Python помогает их автоматизировать.

Расчёт бюджета

Предположим, у нас есть данные о доходах и расходах компании за год, записанные в CSV-файле. Наша задача – рассчитать чистую прибыль по месяцам и выявить, где можно сократить расходы. Сначала мы загружаем данные с помощью pandas:

```
import pandas as pd
# Загрузка данных
d = pd.read_csv('budget.csv')
# Расчёт чистой прибыли
```

```
d['profit'] = d['income'] - d['expenses']  
# Группировка по месяцам  
monthly_profit = d.groupby('month')['profit'].sum()
```

Теперь у нас есть таблица с прибылью по месяцам. Чтобы понять, какие категории расходов самые затратные, можно сделать так:

```
# Анализ расходов по категориям  
expense_by_category = d.groupby('category')['expenses'].sum()  
print(expense_by_category)
```

Допустим, мы видим, что маркетинг съедает слишком много бюджета. Чтобы убедить руководство, построим график с помощью matplotlib:

```
import matplotlib.pyplot as plt  
  
expense_by_category.plot(kind='bar')  
plt.title('Расходы по категориям')  
plt.xlabel('Категория')  
plt.ylabel('Сумма, руб.')  
plt.show()
```

Этот код создаёт столбчатую диаграмму, которая сразу показывает, куда уходят деньги. Всё это занимает пару минут, а результаты можно использовать для обсуждения бюджета.

Анализ инвестиционного портфеля

Теперь представим, что у нас есть портфель из нескольких акций, и мы хотим оценить его доходность и риск. У нас есть данные о ценах акций за год. Сначала загрузим их и рассчитаем дневную доходность:

```
# Загрузка данных о ценах акций  
prices = pd.read_csv('stocks.csv', index_col='date', parse_dates=True)  
# Расчёт дневной доходности  
returns = prices.pct_change().dropna()
```

Теперь используем numpy, чтобы рассчитать среднюю доходность и риск:

```
import numpy as np  
# Средняя доходность и стандартное отклонение  
mean_returns = returns.mean()  
std_returns = returns.std()  
# Коэффициент Шарпа (предполагаем безрисковую ставку 0 для простоты)  
sharpe_ratio = mean_returns / std_returns * np.sqrt(252) # 252 торговых дня в году  
print(sharpe_ratio)
```

Чтобы показать, как активы связаны между собой, построим тепловую карту корреляций:

```
# Корреляционная матрица  
correlation = returns.corr()  
# Визуализация
```

```
plt.figure(figsize=(8, 6))
plt.imshow(correlation, cmap='coolwarm', interpolation='none')
plt.colorbar()
plt.xticks(range(len(correlation)), correlation.columns, rotation=45)
plt.yticks(range(len(correlation)), correlation.columns)
plt.title('Корреляция активов')
plt.show()
```

Эта карта помогает понять, какие активы движутся вместе, а какие диверсифицируют портфель. Такой анализ может стать основой для корректировки инвестиционной стратегии.

Прогнозирование финансовых показателей

Иногда нужно заглянуть в будущее, например, спрогнозировать доходы на основе прошлых данных. Для простоты используем линейную регрессию из библиотеки scikit-learn:

```
from sklearn.linear_model import LinearRegression
# Подготовка данных
d = pd.read_csv('sales.csv')
X = d['month'].values.reshape(-1, 1) # Месяцы
y = d['sales'].values # Продажи
# Обучение модели
model = LinearRegression()
model.fit(X, y)
# Прогноз на следующий месяц
next_month = np.array([[13]])
forecast = model.predict(next_month)
print(f"Прогноз продаж: {forecast[0]:.2f}")
```

Этот код даёт базовый прогноз, который можно визуализировать, чтобы показать тренд:

```
plt.scatter(X, y, label='Исторические данные')
plt.plot(X, model.predict(X), color='red', label='Прогноз')
plt.title('Прогноз продаж')
plt.xlabel('Месяц')
plt.ylabel('Продажи, руб.')
plt.legend()
plt.show()
```

Это простой пример, но он показывает, как Python может помочь предсказать будущее на основе данных, не углубляясь в сложную математику.

Стоит отметить, что представленная модель линейной регрессии является лишь отправной точкой для финансового прогнозирования. На практике финансовые временные ряды часто демонстрируют нелинейную динамику и сезонность. Для повышения точности прогноза можно использовать более сложные модели, такие как ARIMA (Auto-Regressive Integrated Moving Average) или Prophet от Facebook, которые также легко реализуются в Python. Например, библиотека statsmodels позволяет с минимальными усилиями построить ARIMA-модель, учитывающую сезонные колебания, что особенно важно при прогнозировании выручки для бизнесов с сезонным характером спроса. При этом Python

сохраняет свое преимущество в виде автоматизации и воспроизводимости анализа даже при усложнении моделей.

Сравнение методов

Чтобы понять, насколько автоматизация с Python эффективна, давайте сравним её с традиционным подходом, например, с использованием Excel. мы провели небольшой эксперимент: взяли задачу расчёта бюджета и попробовали решить её двумя способами.

В Excel мы вручную загрузили данные, создала формулы для прибыли и построила диаграмму. Это заняло около 20 минут, и каждый раз, когда данные обновлялись, приходилось повторять часть работы. В Python тот же анализ занял 5 минут на написание кода, а повторный расчёт с новыми данными – секунды. Код можно сохранить и использовать снова, что особенно удобно для регулярных отчётов.

Ещё одно преимущество Python – это масштабируемость. Если данных станет больше, Excel начнёт тормозить, а Python справится без проблем благодаря оптимизированным библиотекам. Плюс, Python позволяет автоматизировать даже сложные задачи, такие как интеграция данных из разных источников или построение прогнозов, что в Excel требует значительных усилий.

Однако у Python есть и свои нюансы. Для новичков освоение языка может занять время, особенно если нет опыта программирования. Но хорошая новость в том, что для базового финансового анализа достаточно нескольких команд, а сообщество Python предлагает множество учебных материалов.

Неоспоримым преимуществом Python является также возможность создания конвейеров данных (data pipelines), которые автоматизируют весь процесс от сбора данных до формирования отчетов. Например, можно написать скрипт, который каждый день загружает актуальные финансовые данные из различных источников, проводит их очистку и трансформацию, выполняет анализ и автоматически отправляет отчет заинтересованным лицам. В условиях современного бизнеса, где решения должны приниматься быстро и на основе актуальной информации, такая автоматизация дает значительное конкурентное преимущество. Excel в подобных сценариях требует значительно больше ручного вмешательства, что увеличивает не только трудозатраты, но и риск человеческих ошибок.

Результаты

Применение Python для автоматизации финансового анализа дало впечатляющие результаты. В примере с бюджетом мы смогли сократить время анализа с часов до минут, а результаты стали более точными благодаря автоматическим расчётам. Анализ портфеля показал, что Python не только упрощает вычисления, но и помогает выявить скрытые закономерности, такие как корреляция активов, которые сложно заметить вручную. Прогнозирование продаж, хотя и было базовым, дало понимание трендов, что может стать отправной точкой для более сложных моделей.

Важно отметить, что автоматизация не просто экономит время – она меняет подход к работе. Аналитик перестаёт быть "калькулятором" и начинает задавать более глубокие вопросы: какие данные ещё можно проанализировать? Какие выводы помогут бизнесу? Python даёт свободу экспериментировать и находить новые решения.

Ключевой результат внедрения Python в финансовый анализ – это повышение качества принимаемых решений за счет более глубокого исследования данных. Когда рутинные операции автоматизированы, аналитики могут уделять больше внимания интерпретации результатов и поиску неочевидных закономерностей. Например, при автоматизированном анализе бюджета можно не просто выявить отклонения от плана, но и исследовать их причины, провести "what-if" анализ различных сценариев и предложить обоснованные рекомендации по оптимизации. Такой подход трансформирует роль финансового аналитика от простого исполнителя к стратегическому партнеру в принятии бизнес-решений.

Заключение

Автоматизация финансового анализа с помощью Python – это не просто модный тренд, а реальный способ сделать работу эффективнее и интереснее. Мы увидели, как с помощью pandas, numpy и matplotlib можно быстро обрабатывать данные, проводить сложные расчёты и представлять результаты так, чтобы они вдохновляли на действия. Примеры с бюджетом, портфелем и прогнозированием показали, что даже базовые инструменты Python открывают большие возможности.

Конечно, автоматизация требует начальных усилий – нужно освоить основы языка и разобраться в библиотеках. Но эти вложения окупаются многократно, когда вы начинаете видеть, как рутинные задачи решаются за секунды, а вы сосредотачиваетесь на том, что действительно важно. В будущем мы планируем углубиться в более сложные методы, такие как машинное обучение для финансовых прогнозов, но уже сейчас Python доказал, что он незаменимый помощник для любого аналитика.

Важно подчеркнуть, что автоматизация финансового анализа с помощью Python – это не только технологический, но и культурный сдвиг в работе финансовых специалистов. Она способствует развитию аналитического мышления и междисциплинарного подхода, объединяющего знания в области финансов и информационных технологий. В образовательном контексте это означает необходимость обновления учебных программ для подготовки специалистов нового типа, владеющих как финансовой теорией, так и инструментами для работы с данными. В свете этих тенденций, представленные в статье примеры могут послужить отправной точкой для тех, кто начинает интегрировать Python в свою финансовую практику, открывая дорогу к более инновационным и эффективным подходам в финансовом анализе.

Список литературы

1. Аркабаев Н. К., Доолотбек кызы Г., Аманбаев С. М. Оптимизация логистических процессов и транспортных задач в условиях динамичной онлайн-торговли // Бюллетень науки и практики. 2024. Т. 10. №1. С. 292-298. <https://doi.org/10.33619/2414-2948/98/35>
2. Аркабаев, Н. К. Оптимизация складского учета с использованием технологий Интернета вещей и платформы.NET / Н. К. Аркабаев, А. С. Орозбаева, Т. А. Наралиев // Вестник Ошского государственного университета. – 2024. – № 4. – С. 150-163. – DOI 10.52754/16948610_2024_4_16. – EDN IORHBN.

3. Борисова, М.Н., Дашин, А.К. (2025). Система автоматизации анализа финансовых результатов предприятия, №1(204), сс. 166-169. <https://scilead.ru/article/7862-sistema-avtomatizatsii-analiza-finansovikh-re>
4. Имаралиев, О. Р. Кайра иштетүү ишканаларында өндүрүмдүүлүктү оптималдаштыруу усулдары / О. Р. Имаралиев, М. О. Какеева, Б. А. Абрасулова // Вестник Ошского государственного университета. – 2025. – No. 1. – P. 153-160. – DOI 10.52754/16948610_2025_1_13. – EDN MIVJQW.
5. Как можно использовать Python для финансов <https://ru.bitdegree.org/rukovodstvo/python-dlya-finansov> (дата обращения: 10.04.2025).
6. Наралиев, Т. А. Аймактардагы кайра иштетүү өндүрүш ишканаларына социалдык-экономикалык анализ / Т. А. Наралиев, О. Р. Имаралиев, К. Б. Бакытбек Кызы // Вестник Ошского государственного университета. – 2021. – Vol. 2, No. 1. – P. 76-87. – DOI 10.52754/16947452_2021_2_1_76. – EDN НКYYXJ.
7. Суханова, И.В. (2021). Проектирование автоматизации анализа финансового состояния компании электросетевого комплекса, № 4(44). сс. 28-39. <https://doi.org/10.34822/1999-7604-2021-4-28-39>
8. Челухина, Н.Ф., Асяева, Э. А. (2021). Цифровые технологии в страховом бизнесе: преимущества и риски. Материалы III Международной научной конференции “Тенденции экономического развития в XXI веке” (сс. 1038-1041). Минск: БГУ
9. VanderPlas, J. (2016). Python Data Science Handbook: Essential Tools for Working with Data, O'Reilly Media. Beijing: O'Reilly.
10. Sharpe, W.F. (1994). The Sharpe Ratio. Journal of Portfolio Management, 21(1), pp. 49-58. <http://dx.doi.org/10.3905/jpm.1994.409501>
11. Stop Using Excel for Data Analytics: Upgrade to Python <https://towardsdatascience.com/stop-using-excel-for-data-analytics-upgrade-to-python-46b5963fb036/> (дата обращения: 10.04.2025).