

УДК 330.4

ИСПОЛЬЗОВАНИЕ ЯЗЫКА ПРОГРАММИРОВАНИЯ PYTHON ДЛЯ АНАЛИЗА ЭКОНОМИЧЕСКИХ ДАННЫХ

И.Ю. Ваславская, Д.М. Лысанов

Набережночелнинский институт (филиал) Казанского (Приволжского) федерального университета, Набережные Челны, email: vaslavskaya@yandex.ru

Аннотация. В статье рассматриваются возможности применения языка программирования Python для обработки и анализа экономических данных. Приводятся методы его использования, включая дескриптивный анализ, очистку и трансформацию данных, построение эконометрических и прогнозных моделей, а также визуализацию результатов. Особое внимание уделено библиотекам – Pandas, NumPy, Statsmodels, Scikit-learn, Matplotlib, Plotly и их роли в реализации аналитических подходов, от классической регрессии до методов машинного обучения и интерактивной визуализации данных.

Ключевые слова: язык Python, экономический анализ, обработка данных; эконометрическое моделирование; визуализация данных.

USING THE PYTHON PROGRAMMING LANGUAGE FOR ECONOMIC DATA ANALYSIS

I.Yu. Vaslavskaya, D.M. Lysanov

Naberezhnye Chelny Institute (branch) of Kazan (Volga Region) Federal University, Naberezhnye Chelny, email: vaslavskaya@yandex.ru

Abstract. The article discusses the possibilities of using the Python programming language for processing and analyzing economic data. Methods of its use are presented, including descriptive analysis, data cleaning and transformation, construction of econometric and forecasting models, as well as visualization of results. Particular attention is paid to libraries – Pandas, NumPy, Statsmodels, Scikit-learn, Matplotlib, Plotly and their role in implementing analytical approaches, from classical regression to machine learning methods and interactive data visualization.

Keywords: Python language, economic analysis, data processing; econometric modeling; machine learning, data visualization.

Дата поступления статьи в редакцию: 07.01.2026

Дата принятия статьи в печать: 20.02.2026

Введение

Экономические данные, отражающие различные экономические процессы и описывающие состояние экономики на разных уровнях, сегодня относят к большим данным. Они необходимы для формирования эмпирической базы при проведении исследований с целью обоснования принимаемых решений в науке, экономике, разных сферах бизнеса, а также в государственном управлении. В результате применения технологий обработки больших данных можно поучать информацию о состоянии объекта данных в разные периоды времени. Исследователи отмечают, что технологии, работающие с большими данными, должны обладать рядом функциональных особенностей, которые позволяют с высокой скоростью обработать возрастающий объем и разнообразие данных [1]. Для этого необходимы инструменты, которые могут быстро и точно обрабатывать огромные массивы данных в процессах проведения экономических исследований, финансового анализа, подготовки прогноза.

Одним из таких инструментов является язык программирования Python, который сегодня успешно применяется для анализа данных, их визуализации, статистического анализа и интерпретации результатов в экономических исследованиях. Проблема исследования заключается в изучении возможностей использования языка Python для решения экономических задач в процессе проведения экономических исследований.

Возможностям использования языка программирования Python для решения экономических задач посвящается значительное количество научных публикаций. Среди них можно выделить статью, размещенную на платформе Хабр, в которой рассматриваются различные подходы к анализу и классификации данных с использованием библиотек Python [2]. Хотя авторы акцен-

тируют внимание на общих методах, они вполне могут быть адаптированы для экономических исследований. Изучение материалов по рассматриваемой проблематике показывает, что анализ данных при помощи языка программирования Python является одним из самых популярных и эффективных способов обработки больших объемов информации [3, 4].

Цель исследования

Целью исследования является изучение возможностей применения языка программирования Python для анализа экономических данных, а также проблем, которые могут возникать на практике при обработке, визуализации данных и моделировании для решения экономических задач.

Материал и методы исследования

К основному достоинству Python следует отнести понятный и простой синтаксис, позволяющий работать с ним разным категориям пользователей, например, аналитикам и исследователям. Кроме того, для Python существует экосистема готовых библиотек, созданных для работы с данными, он имеет открытый исходный код, что обеспечивает прозрачность, воспроизводимость результатов и возможность совершенствования языка [5]. Использование языка программирования Python для анализа экономической информации в сферах экономики и финансов позволяет решать задачи обработки данных, их визуализации, статистического анализа и интерпретации результатов.

Процесс обработки информации начинается с простого описательного анализа данных, с помощью которого можно быстро получить базовую статистику, оценить распределение данных, выявить аномалии или пропуски. Специальные библиотеки Python предоставляют широкий набор инструментов для обработки огромных массивов информации [3]. Так, например, для прогнозирования Python выполняет ARIMA анализ, может использовать методы машинного обучения, градиентный спуск и нейросети. Библиотеки позволяют сравнивать подходы, проверять устойчивость прогнозов, и подгонять модели под конкретную задачу. Это особенно ценно, когда экономические временные ряды имеют выбросы и ведут себя нелинейно [6].

В Python можно проводить простой эконометрический анализ с помощью линейной и нелинейной регрессии, а также строить сложные модели с разнородными переменными или бинарным выбором. Также Python хорошо подходит для визуализации данных, учета функции потерь или вычисления стандартных ошибок, построения как обычных графиков-гистограмм, диаграмм рассеяния, временных рядов, – так и интерактивных дашбордов [7, 8].

Таким образом, Python обладает широким набором инструментов, которые используются для обработки данных, а также при работе с таблицами для прочтения, группировки и преобразования данных, что особенно важно в случае пропуска, дублирования данных, или их неверных форматов. С помощью библиотеки для языка программирования Python NumPy можно работать с многомерными массивами и матрицами, оперативно решать системы линейных уравнений, задачи сложения и умножения данных, транспонирования матриц.

Библиотеки для визуализации данных в Python Matplotlib и Seaborn позволяют выполнить построение простых, наглядных статистических графиков, графиков регрессии. Для сложной эконометрики применяется Statsmodels, которая помогает строить линейные регрессии с диагностическими параметрами – проверкой на гетероскедастичность, автокорреляцию, работать с временными рядами – от простых до сложных моделей.

Библиотека на языке Python с открытым исходным кодом Scikit-learn используется для машинного обучения – от регрессий и классификации до кластеризации. В экономике его применяют для прогнозирования спроса, оценки кредитного риска, поиска аномалий данных или сегментирования клиентов с помощью встроенных методов кросс-валидации и подбора параметров.

Библиотеки Plotly и Vokeh позволяют создавать интерактивные дашборды, «живые» графики, которые можно приближать, на которые можно наводить мышку, чтобы увидеть детали, и даже фильтровать данные.

Программный интерфейс приложений API (Application Programming Interface) выполняет роль посредника, с помощью которого разные программы могут взаимодействовать и обмениваться данными, обеспечивая их передачу данных и выполнение операций. Например, для сбора данных через API можно напрямую подключиться к различным открытым источникам, содержащим сотни макроэкономических индикаторов. При этом отсутствует ручной экспорт-импорт данных, что сводит ошибки к минимуму [9, 10].

С помощью программной библиотеки на языке Python Pandas реализуются такие ключевые этапы экономического анализа, как сбор данных, их обработка и анализ. Для структурирования информации данные представляются в табличном формате, чаще всего CSV и Excel. В библиотеке Pandas есть функции `read_csv()` и `read_excel()`, которые разбираются в структуре таблицы, определяют, где и какие данные, находят пропуски и даже угадывают кодировку. При этом пользователь может применить различные настройки – задать разделитель, указать, где заголовок, или выбрать нужный лист в Excel.

Через API можно запрашивать макроэкономические показатели – инфляцию, безработицу, ставки – прямо в коде, вообще без скачивания файлов. Банковские клиенты работают по тому же принципу. Существуют готовые библиотеки, которые умеют превращать запросы к сайту Росстата в структурированные данные. В случаях, когда данных нет ни в файлах, ни через API, применяется веб-скрапинг, то есть автоматическое извлечение информации с сайтов. Для простых, статичных сайтов, где все содержимое загружается сразу, хватает связки `requests` и `BeautifulSoup`. Первая библиотека скачивает HTML-страницу, а вторая помогает ее распарсить и получить нужные данные. Но существуют сайты, которые подгружают контент динамически с помощью JavaScript. В таких случаях потребуется `Selenium`, который имитирует браузер и может нажать на кнопки и подождать, пока таблица загрузится, обойти простейшую защиту от парсинга.

После загрузки данных Python превращает их в один формат – DataFrame, который позволяет строить целые конвейеры обработки. Например, если входные данные обновились, то автоматически пересчитываются все показатели и графики.

При подготовке данных для экономического анализа часто возникает проблема, связанная с пропущенными значениями. Они появляются по разным причинам, например, при техническом сбое при сборе данных или в случае, когда показатель был недоступен. Для решения этой проблемы можно использовать функции, которые проверяют значения на `NaN` (то есть на «пустоту»), а дальше уже выбрать, что делать: удалить такие строки, заполнить их средним, медианой или предыдущим значением. Можно восстановить данные через регрессию или метод *k*-ближайших соседей [11].

Еще одна возможная проблема – правильные типы данных. Иногда числа из-за пробелов или значков валюты считываются как текст, или даты остаются текстом, и тогда ни тренды не построить, ни лаги не посчитать. Функция `to_datetime()` распознает почти любой формат даты, а какой-нибудь значок рубля или доллара легко убрать парой строк кода и превратить все в числа [3].

Для решения проблемы дубликатов и выбросов, которая возникает при объединении данных из разных мест, и появляются полные копии строк, используется одна простая команда `drop_duplicates()`. Для работы с частичными дублями нужны специальные методы сравнения строк. Выбросы нельзя просто исключить, считая ошибкой. В экономике аномальные цифры часто отражают реальные вещи: кризисы, крупные сделки, какие-то структурные сдвиги в экономике. Выбросы можно прологарифмировать, чтобы сгладить их влияние, или заменить на какое-то пороговое значение.

Нормализация и стандартизация становятся особенно важны, когда переменные представлены в разных масштабах. Нормализация «сжимает» все значения в диапазон от 0 до 1, а стандартизация приводит все к среднему нулю и единичной дисперсии. Делать это целесообразно уже после разделения данных на обучающую и тестовую выборки, чтобы не было потери информации.

Расчет ключевых экономических показателей – это основа для анализа экономики всей страны или исследования деятельности отдельной компании. Самые ходовые метрики, например, темпы роста, инфляция или ВВП на душу населения, обычно считают по временным рядам или панельным данным. Графики помогают увидеть закономерности, которые в цифрах просто не заметны. Например, можно наложить на график линейный тренд и увидеть, как замедляется рост доходов, пока инфляция растет. Или построить гистограмму и понять, как распределены данные: есть ли асимметрия или какие-то аномальные значения [12].

Для формализации связи между простым описанием данных и их моделированием применяется корреляционный анализ, который может показать, насколько сильно связаны две переменные, например, безработица и доходы людей. Это отражает такой инструмент как коэф-

фициент Пирсона, однако он плохо работает с выбросами и нелинейными связями. В таких случаях применяется ранговая корреляция Спирмена, которая достаточно устойчива к выбросам и нестандартным значениям.

Регрессионные модели позволяют в цифрах оценить, как одни переменные влияют на другие, причем с учетом разных факторов. Модель задается формулой, которая очень похожа на обычную математическую запись. Программа автоматически считает все стандартные ошибки вместе с доверительными интервалами и вспомогательными параметрами, которые используются для диагностики.

На каждом этапе проверяются гипотезы. Для проверки значимости отдельных коэффициентов есть t-статистика. Для оценки важности группы факторов применяется F-тест. Для проверки ограничений и сравнения моделей применяются тесты Вальда, множителей Лагранжа и отношения правдоподобия [13]. Чтобы окончательно убедиться, что с моделью все в порядке, анализируются остатки. Их график показывает, не пропущены ли структурные сдвиги, а диаграмма «предсказания-остатки» укажет нелинейность [4].

В Python можно строить как точечные прогнозы, так и интервальные, которые показывают неопределенность данных. А для временных рядов есть специальные методы ARIMA и ее производных, которые описывают динамику через прошлые значения и случайные отклонения [3].

Результаты исследования

Результаты исследования возможностей применения языка программирования Python имеют ценность для развития теоретических подходов к проведению прикладных экономических исследований, а также при выстраивании оптимальной архитектуры больших данных в сфере экономики и управления. Для примера рассмотрим, как с помощью языка программирования Python решается конкретная практическая задача по экономическому анализу. Проанализируем, как в России за последние годы были связаны динамика ВВП и уровень безработицы на основе открытых данных Росстата за 2004–2023 годы: реальный ВВП (в % к предыдущему году) и уровень безработицы (в % от экономически активного населения). Выделим основные этапы алгоритма решения поставленной задачи.

1. Подготовительный шаг – загрузка специализированных библиотек Python, которые предоставляют инструменты для различных этапов анализа данных, статистического моделирования и визуализации.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.api as sm
```

2. Создание структурированного набора данных для анализа, который формирует таблицу с временным рядом макроэкономических показателей и подготавливает ее к анализу во временном контексте.

```
data = {
    'year': list(range(2004, 2024)),
    'gdp_growth_real': [
        7.2, 6.4, 8.2, 8.5, 5.7, 2.5, 4.5, 5.2, 3.7, 1.8,
        0.7, -2.0, 2.5, 1.8, 2.3, 2.2, -3.5, 4.7, 3.5, 3.6
    ],
    'unemployment': [
        8.1, 7.2, 6.3, 5.5, 5.8, 8.4, 7.3, 5.9, 5.5, 5.5,
        5.3, 5.4, 5.4, 5.2, 4.8, 4.6, 5.3, 4.9, 3.9, 3.2
    ]
}
df = pd.DataFrame(data)
df['year'] = pd.to_datetime(df['year'], format='%Y')
df.set_index('year', inplace=True)
```

3. Построение комбинированного графика с двумя осями Y для одновременной визуализации и сравнения динамики двух ключевых макроэкономических показателей – роста ВВП и уровня безработицы.

```
plt.rcParams.update({'font.size': 11})
plt.style.use('seaborn-v0_8-whitegrid')
fig, ax1 = plt.subplots(figsize=(12, 5))
color = 'tab:green'
ax1.set_xlabel('Год')
ax1.set_ylabel('Темп роста ВВП, % (год к году)', color=color)
ax1.plot(df.index, df['gdp_growth_real'], color=color, marker='o',
label='Реальный ВВП рост', linewidth=2)
ax1.tick_params(axis='y', labelcolor=color)
ax1.axhline(0, color='gray', linestyle='--', linewidth=0.8)
ax2 = ax1.twinx()
color = 'tab:purple'
ax2.set_ylabel('Безработица, %', color=color)
ax2.plot(df.index, df['unemployment'], color=color, marker='s',
label='Безработица', linewidth=2)
ax2.tick_params(axis='y', labelcolor=color)
fig.suptitle('Динамика реального ВВП и безработицы в РФ (2004-2023)')
fig.tight_layout()
plt.show()
```

4. Количественный статистический анализ взаимосвязи между двумя макроэкономическими показателями: темпом роста реального ВВП и уровнем безработицы. Он позволяет перейти от визуальной оценки графиков к точным численным оценкам силы и характера этой связи.

```
corr = df['gdp_growth_real'].corr(df['unemployment'])
print(f"\nКоэффициент корреляции Пирсона: {corr:.3f}")
X = sm.add_constant(df['gdp_growth_real'])
y = df['unemployment']
model = sm.OLS(y, X).fit()
print(«\nРезультаты регрессии (безработица ~ темп роста ВВП):»)
print(model.summary().tables[1])
```

На следующем этапе строится график рассеяния с линией регрессии, который визуализирует результаты проведенного регрессионного анализа. Этот график служит для прямой визуальной оценки качества модели, связи между переменными и позволяет увидеть, насколько хорошо теоретическая модель соответствует реальным данным.

```
plt.figure(figsize=(10, 4))
plt.scatter(df['gdp_growth_real'], df['unemployment'], color='navy',
label='Наблюдения')
plt.plot(df['gdp_growth_real'], model.fittedvalues, color='red', label='Линия
регрессии', linewidth=2)
plt.xlabel('Темп роста ВВП, %')
plt.ylabel('Безработица, %')
plt.title('Зависимость уровня безработицы от экономического роста (РФ,
2004-2023)')
plt.legend()
plt.grid(True)
plt.show()
```

5. Тест на наличие автокорреляции остатков построенной регрессионной модели с помощью статистики Дарбина–Уотсона. Это важный этап диагностики модели при работе с данными, представляющими собой временные ряды.

```
dw = sm.stats.durbin_watson(model.resid)
print(f»\nСтатистика Дарбина–Уотсона: {dw:.2f}»)
```

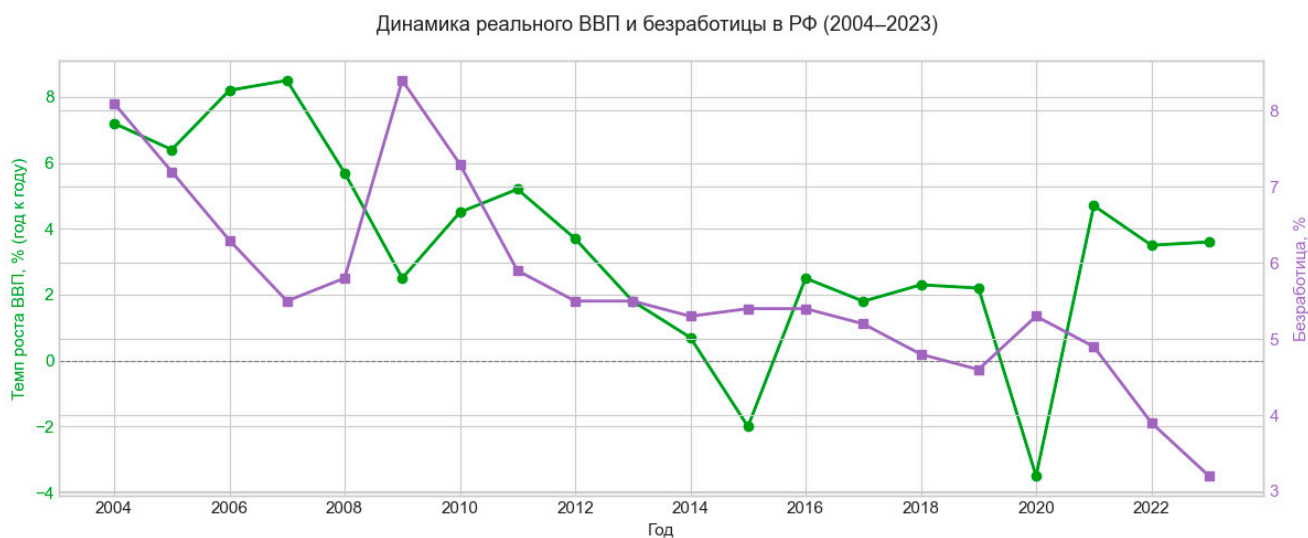


Рис. 1. Динамика ВВП и безработицы в РФ

Таблица 1

Регрессионный анализ взаимосвязи между темпом роста ВВП и уровнем безработицы

	Coef	std err	t	P> t	[0.025	0.975]
const	5.2280	0.431	12.125	0.000	4.322	6.134
gdp_growth_real	0.1286	0.094	1.365	0.189	-0.069	0.327

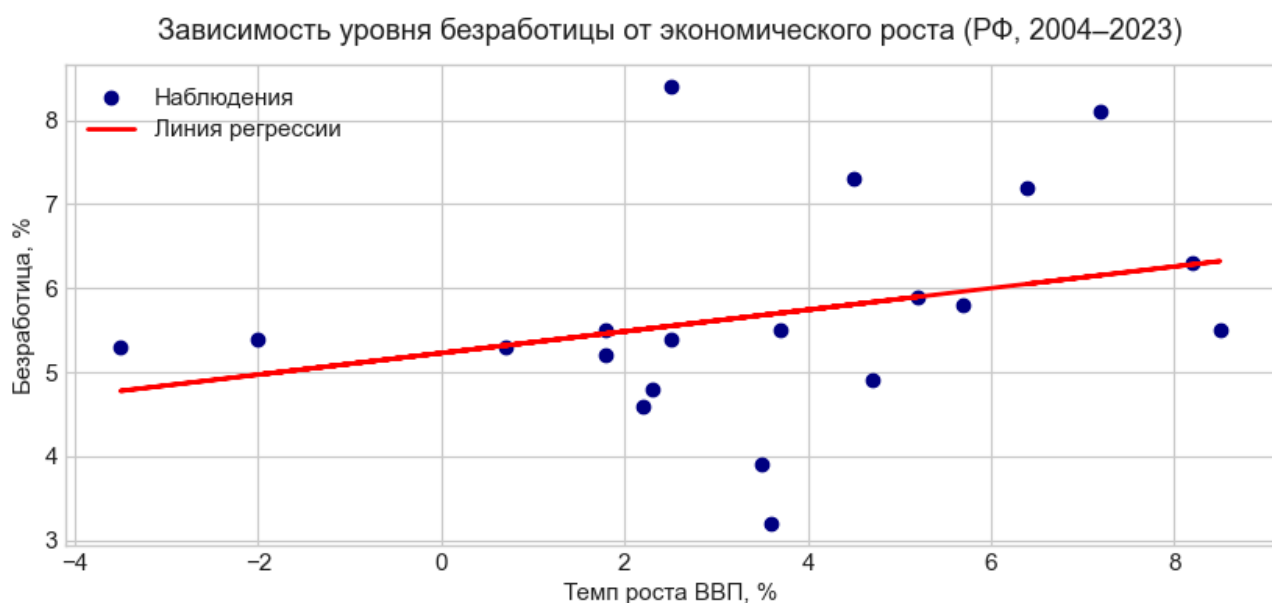


Рис. 2. Зависимость уровня безработицы от экономического роста в РФ

После выполнения программы получим следующие результаты:

1. На основе динамики реального ВВП и безработицы в РФ за период 2003– 2024 гг. построены графики зависимостей (рис. 1).
2. Рассчитан коэффициент корреляции Пирсона = 0,306, полученное значение которого отражает линейную корреляцию между двумя наборами исходных данных, не учитывая другие виды взаимосвязей.
3. Выполнен регрессионный анализ взаимосвязи между двумя макроэкономическими показателями: темпом роста ВВП и уровнем безработицы (табл. 1).
4. Построена визуализация статистической модели в виде линейной регрессии (рис. 2), она позволяет оценить точность предсказания уровня безработицы в РФ, зная темпы роста экономики.
5. Рассчитано значение статистики Дарбина–Уотсона = 0,79, означающее положительную автокорреляцию, т.е. зависимость уровня безработицы в России от темпов экономического роста за рассматриваемый период.

Заключение

Таким образом, широкие возможности языка программирования Python позволяют использовать его как эффективный инструмент для анализа экономических данных. Необходимо также учитывать, что экосистема Python растет быстрыми темпами, библиотеки постоянно обновляются, появляются новые инструменты для работы с большими данными, машинного обучения, интерактивной аналитики.

Применение языка программирования Python для решения задач экономического анализа и прогнозирования на основе классической эконометрики и анализа текстов в современных быстро меняющихся условиях позволяет усовершенствовать документацию, повысить надежность библиотек и находить ответы на самые сложные вопросы.

Python – это не просто язык программирования, а целая среда для исследований, где серьезный научный подход сочетается с эффективными инструментами исследования. И чем сложнее становятся экономические задачи и чем больше имеется данных, тем ценнее результаты применения Python.

Литература

1. Сенько А. Работа с BigData в облаках. Обработка и хранение данных с примерами из Microsoft Azure. СПб.: Питер, 2019. 448 с.
2. Анализ данных с использованием Python. [Электронный ресурс]. URL: <https://habr.com/ru/articles/353050/> (дата обращения: 16.11.2025).
3. Маккинни У. Python и анализ данных / пер. с англ. А. В. Логунова. М.: ДМК Пресс, 2022. 544 с.
4. Фарайдунов О.К., Амонбекова З.А. Анализ данных в среде языка программирования PYTHON для решения экономических задач // Финансово-экономический вестник. 2024. № 4.1(43). С. 91-98.
5. Pandas documentation. [Электронный ресурс]. URL: <https://pandas.pydata.org/docs/> (дата обращения: 10.11.2025).
6. Хантессаридис А., Шевченко А.В. Эконометрика с применением Python: учеб.-метод. пособие. М.: Издательский дом НИУ ВШЭ, 2024. 218 с.
7. Айвазян С.А., Мхитарян В.С. Прикладная статистика и основы эконометрики. 2-е изд., перераб. и доп. М.: ЮНИТИ-ДАНА, 2020. 959 с.
8. Высшая школа экономики. Анализ данных на Python. [Электронный ресурс]. URL: <https://www.coursera.org/learn/python-for-data-analysis> (дата обращения: 25.11.2025).
9. Федеральная служба государственной статистики. [Электронный ресурс]. URL: <https://rosstat.gov.ru/opendata> (дата обращения: 21.11.2025).
10. Центральный банк Российской Федерации. Открытые данные и API. [Электронный ресурс]. URL: <https://www.cbr.ru/development/> (дата обращения: 21.11.2025).
11. Балабанов И.Т., Балабанов А.И. Анализ данных в экономике: учеб. пособие. СПб.: Питер, 2023. 352 с.
12. Economic Data Analysis with Python. [Электронный ресурс]. URL: <https://github.com/ru-econ-data/python-tutorials> (дата обращения: 28.11.2025).
13. Statsmodels documentation. [Электронный ресурс]. URL: <https://www.statsmodels.org/stable/index.html> (дата обращения: 27.11.2025).