

ИНФОРМАТИКА И ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ/INFORMATICS AND INFORMATION PROCESSES

DOI: <https://doi.org/10.60797/IRJ.2026.164.70>

МЕТОДЫ ВИЗУАЛЬНОГО ПРЕДСТАВЛЕНИЯ БОЛЬШИХ ДАННЫХ С ИСПОЛЬЗОВАНИЕМ PLOTLY И MATPLOTLIB

Научная статья

Нуриев М.Г.^{1,*}, Мангушева А.Р.², Махфуд Б.А.³, Зязюля И.С.⁴¹ORCID : 0009-0003-0741-1734;^{1, 3, 4} Казанский национальный исследовательский технический университет им. А.Н. Туполева – КАИ, Казань, Российская Федерация² Казанский национальный исследовательский технологический университет, Казань, Российская Федерация

* Корреспондирующий автор (mgnuriev[at]kai.ru)

Аннотация

В статье рассматривается комплексный подход к визуализации статистических данных в среде Jupyter Notebook. Особое внимание уделяется анализу современных методов графического представления информации, выбору оптимальных инструментов и библиотек, а также когнитивным особенностям восприятия визуализаций. Показаны преимущества интеграции Plotly и Matplotlib для создания как статических, так и интерактивных графиков, рассмотрены вопросы оптимизации работы с большими массивами данных и применения параллельных вычислений. Продемонстрирована практическая значимость использования Jupyter Notebook для анализа, документирования и представления данных в образовательной, научной и бизнес-среде. Перспективы исследования связаны с внедрением методов машинного обучения и облачных сервисов для автоматизации выбора оптимальных визуализаций и расширения возможностей коллективной работы.

Ключевые слова: визуализация данных, Jupyter Notebook, Plotly, Matplotlib, большие данные, статистический анализ, параллельные вычисления, оптимизация, базы данных, аналитика.

METHODS FOR VISUALISING BIG DATA USING PLOTLY AND MATPLOTLIB

Research article

Nuriev M.G.^{1,*}, Mangusheva A.R.², Makhfud B.A.³, Zyazyulya I.S.⁴¹ORCID : 0009-0003-0741-1734;^{1, 3, 4} Kazan National Research Technical University named after A.N. Tupolev – KAI, Kazan, Russian Federation² Kazan National Research Technological University, Kazan, Russian Federation

* Corresponding author (mgnuriev[at]kai.ru)

Abstract

The article examines a complex approach to visualising statistical data in the Jupyter Notebook environment. Particular attention is paid to analysing modern methods of graphical representation of information, selecting optimal tools and libraries, as well as cognitive traits of perception of visualisations. The advantages of integrating Plotly and Matplotlib for creating both static and interactive graphs are demonstrated, and issues of optimising work with large data arrays and the application of parallel computing are discussed. The practical significance of using Jupyter Notebook for analysing, documenting, and presenting data in educational, scientific, and business environments is shown. Future research prospects are related to the implementation of machine learning methods and cloud services to automate the selection of optimal visualisations and expand the possibilities for collaborative work.

Keywords: data visualisation, Jupyter Notebook, Plotly, Matplotlib, big data, statistical analysis, parallel computing, optimisation, databases, analytics.

Введение

Современный цифровой мир характеризуется беспрецедентным ростом объемов генерируемых данных, что создает актуальную потребность в разработке эффективных методов их визуального представления. Грамотная визуализация статистических данных не только помогает выявить скрытые закономерности и взаимосвязи, но и значительно облегчает процесс аналитики, делая сложную информацию доступной для восприятия различными категориями пользователей. Однако многие существующие подходы к визуализации зачастую не учитывают когнитивные особенности человеческого восприятия информации, что может приводить к серьезным искажениям передаваемой информации и принятию ошибочных решений на ее основе.

Основной целью представленного исследования является разработка комплексного методологического подхода к визуальному представлению статистических данных в среде Jupyter Notebook. Данный подход включает многогранный анализ существующих методов визуализации с точки зрения их эффективности и наглядности, разработку оптимизированных цветовых схем и композиционных решений, создание интерактивных элементов для углубленного анализа данных, а также формулирование практических рекомендаций по выбору типа визуализации в зависимости от специфики данных и решаемых аналитических задач.

Jupyter Notebook представляет собой современную интерактивную веб-среду для работы с кодом и данными, которая открывает принципиально новые возможности для создания динамических и интерактивных визуализаций [1],

[2]. Его уникальность заключается в эффективном сочетании вычислительной мощности языков программирования (преимущественно Python) с интуитивно понятным интерфейсом, что делает данную платформу идеальным решением для исследовательской работы с данными различной сложности и объема.

Анализ современных методов визуализации статистических данных

Визуальное представление данных играет ключевую роль в современной аналитике, существенно облегчая процесс понимания и анализа сложных информационных массивов. Среди наиболее значимых преимуществ грамотной визуализации можно выделить несколько фундаментальных аспектов. Во-первых, она значительно упрощает восприятие сложной информации, преобразуя абстрактные числовые данные в наглядные графические образы [3], [4]. Во-вторых, визуализация позволяет мгновенно выявлять важные тенденции и закономерности, которые часто остаются незамеченными при работе с табличными представлениями данных. В-третьих, она обеспечивает эффективную поддержку процесса принятия решений. В-четвертых, качественная визуализация повышает вовлеченность аудитории, делая презентации и отчеты более интересными и запоминающимися. И наконец, она кардинально упрощает анализ больших данных, позволяя выявлять скрытые паттерны в огромных массивах информации.

Ярким примером эффективной визуализации может служить анализ динамики акций компании NVIDIA Corporation. Графическое представление данных позволяет не только оценить общий рост компании за продолжительный период времени, но и идентифицировать ключевые моменты значительных изменений. Например, резкий скачок стоимости акций в конце 2023 года наглядно демонстрирует реакцию рынка на увеличение спроса на вычислительные мощности, связанные с развитием технологий искусственного интеллекта. Такая визуализация предоставляет заинтересованным сторонам — инвесторам, аналитикам и другим участникам рынка — возможность моментально оценить текущее состояние компании и перспективы ее развития, делая обоснованные выводы на основе одного графика (рис. 1).

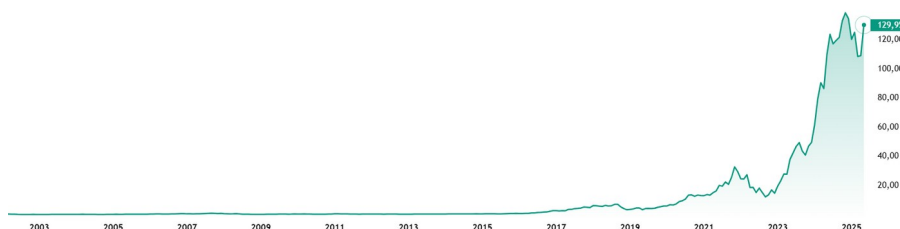


Рисунок 1 - Пост акций компании NVIDIA Corporation
DOI: <https://doi.org/10.60797/IRJ.2026.164.70.1>

Проведенный сравнительный анализ инструментов для визуализации данных демонстрирует значительные преимущества Jupyter Notebook перед такими популярными аналогами, как Excel, Google Colab, Apache Zeppelin, Deepnote и Databricks Notebook [5], [6]. Хотя каждый из этих инструментов обладает определенными достоинствами в специфических сценариях использования, именно Jupyter Notebook предлагает оптимальное сочетание функциональности, гибкости и доступности для широкого спектра задач анализа данных.

Традиционно популярный Excel, несмотря на свою распространенность и привычный интерфейс, показывает серьезные ограничения при работе с большими объемами данных. При превышении порога в 600000 записей проявляются существенные проблемы: экспоненциальный рост времени обработки операций, частые сбои и зависания при выполнении фильтрации и агрегации данных, а также принципиальная невозможность построения сложных многоуровневых визуализаций. Эти ограничения делают Excel малоприспособленным для современной работы с большими данными, где требуются производительные и надежные инструменты анализа.

Принципы эффективной визуализации в Jupyter Notebook

Разработка эффективных визуализаций в Jupyter Notebook требует соблюдения ряда фундаментальных принципов, основанных на особенностях человеческого восприятия и когнитивной психологии. Логика визуализации должна начинаться с четкого определения цели — какой основной месседж необходимо донести до аудитории. Данные должны быть организованы таким образом, чтобы подчеркивать ключевые аспекты и выявлять наиболее важные закономерности, обеспечивая последовательное и логичное повествование [7].

Выбор цветовой палитры является критически важным аспектом эффективной визуализации. Цвета оказывают значительное влияние на восприятие информации и должны использоваться осмысленно и последовательно. Общепринятой практикой является использование зеленого цвета для обозначения положительных аспектов и красного — для отрицательных, однако важно учитывать культурные особенности целевой аудитории. Необходимо придерживаться единой цветовой схемы во всех элементах визуализации, чтобы избежать путаницы и обеспечить целостность восприятия. Правильное использование цвета может значительно улучшить наглядность и эффективность представления информации, в то время как неправильный выбор цветов может привести к серьезным искажениям восприятия и ошибочным выводам [8], [9].

Размер и читаемость текстовых элементов играют критическую роль в эффективности визуализации. При подготовке презентационных материалов необходимо обеспечивать максимальную читаемость текста для всех участников, включая тех, кто находится на значительном удалении от экрана. Практика показывает, что использование

слишком мелкого шрифта приводит к затруднениям в восприятии информации, в то время как крупный шрифт редко вызывает жалобы. Это особенно важно в академической среде, где презентации часто содержат специальные термины, формулы и статистические данные. Рекомендуется проводить предварительное тестирование читаемости материалов в условиях, максимально приближенных к реальным, чтобы обеспечить комфортное восприятие информации всеми читателями (рис. 2).

Размер шрифта	Ширина экрана		
	2 м	2,5 м	3,5 м
16 пт. строчные буквы	4	5	6
18 пт. строчные буквы	6	7	8
20 пт. строчные буквы	9	11	13
22 пт. строчные буквы	11	12	15
24 пт. строчные буквы	13	15	18
30 пт. строчные буквы	15	18	21
32 пт. строчные буквы	19	21	25

Рисунок 2 - Читаемость текста от расстояния
DOI: <https://doi.org/10.60797/IRJ.2026.164.70.2>

Принцип максимальной простоты предполагает устранение всех отвлекающих и не несущих смысловой нагрузки элементов. Следует избегать излишних украшений, сложных визуальных эффектов и декоративных деталей, которые не добавляют ценности содержанию. Часто даже простые двумерные графики позволяют эффективно передавать информацию, облегчая ее восприятие и обработку. Примером излишне усложненной визуализации может служить круговая диаграмма с трехмерным оформлением и неудачно подобранной цветовой палитрой, где 3D-эффекты [10], [11] искажают визуальное восприятие размеров сегментов, а недостаточная контрастность цветов затрудняет интерпретацию данных (рис. 3,4).

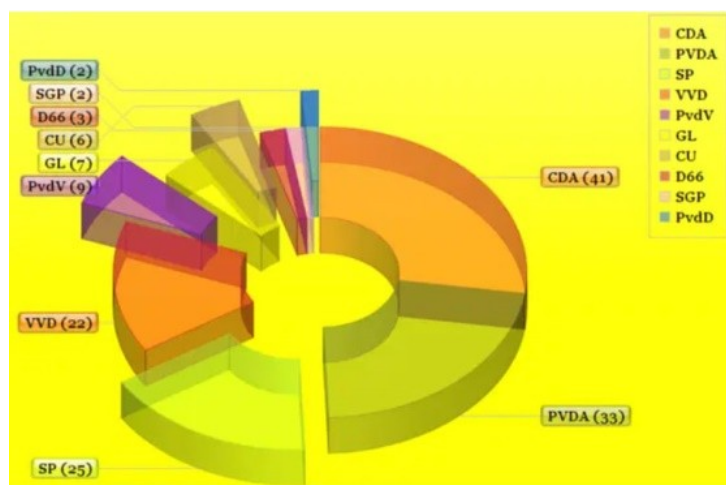


Рисунок 3 - Пример сложной визуализации
DOI: <https://doi.org/10.60797/IRJ.2026.164.70.3>

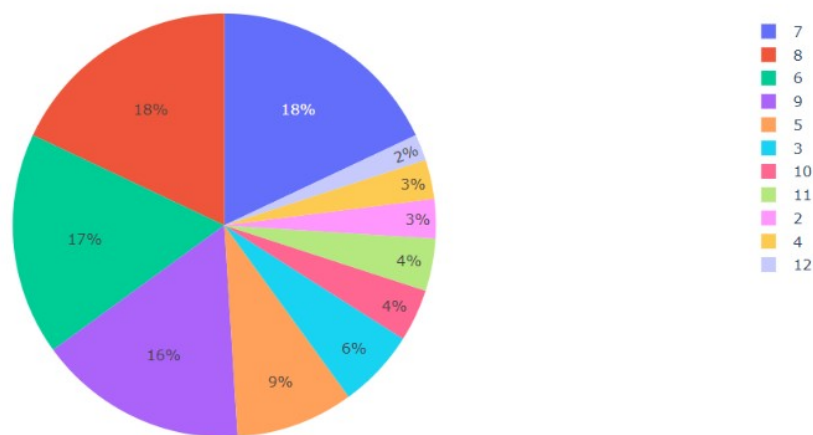


Рисунок 4 - Пример простой визуализации
DOI: <https://doi.org/10.60797/IRJ.2026.164.70.4>

Функционал Jupyter Notebook: библиотеки Plotly и Matplotlib

Процесс работы в Jupyter Notebook начинается с загрузки данных из различных источников: CSV-файлов, таблиц Excel, баз данных SQL/PostgreSQL или веб-API. После загрузки данные проходят этапы очистки, преобразования и анализа с использованием специализированных библиотек, а конечным этапом является визуализация результатов для наглядного представления выявленных закономерностей [12].

Библиотека Plotly — это современный инструмент для интерактивной визуализации данных, который гармонично интегрируется в аналитический рабочий процесс Jupyter. Ключевое преимущество — возможность динамического взаимодействия с графиками непосредственно в ячейках, что значительно ускоряет исследовательский процесс. В отличие от статических решений, Plotly позволяет менять масштаб, скрывать/показывать элементы и выделять участки графиков, что особенно ценно при работе со сложными статистическими зависимостями (листинг 1, рис. 5).

Листинг 1. Код для построения линейного графика

```
import numpy as np
import plotly.express as px

x = np.arange(0, 5, 0.1)

def f(x):
    return x**2

px.scatter(x=x, y=f(x)).show()
```

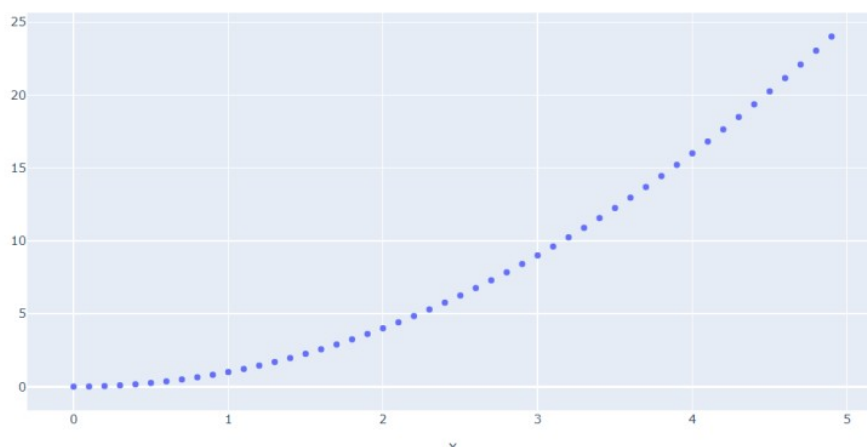


Рисунок 5 - Линейный график
DOI: <https://doi.org/10.60797/IRJ.2026.164.70.5>

Минималистичная визуализация служит основой для дальнейшего расширения функциональности. Ключевое преимущество Plotly — реализация интерактивных элементов, особенно механизма отображения контекстной информации при наведении курсора (листинг 2, рис. 6).

Листинг 2. Код для построения сложного линейного графика

```
import plotly.graph_objects as go

fig = go.Figure()

fig.add_trace(go.Scatter(x=x, y=f(x), mode='lines+markers', name='f(x)=x2'))
fig.add_trace(go.Scatter(x=x, y=g(x), mode='markers', name='$g(x)=x$'))

fig.update_layout(legend_orientation='h', legend=dict(x=.5,
xanchor="center"), margin=dict(l=0, r=0, t=0, b=0)) fig.update_traces(hoverinfo='x+y', r=0, t=0, b=0)
```

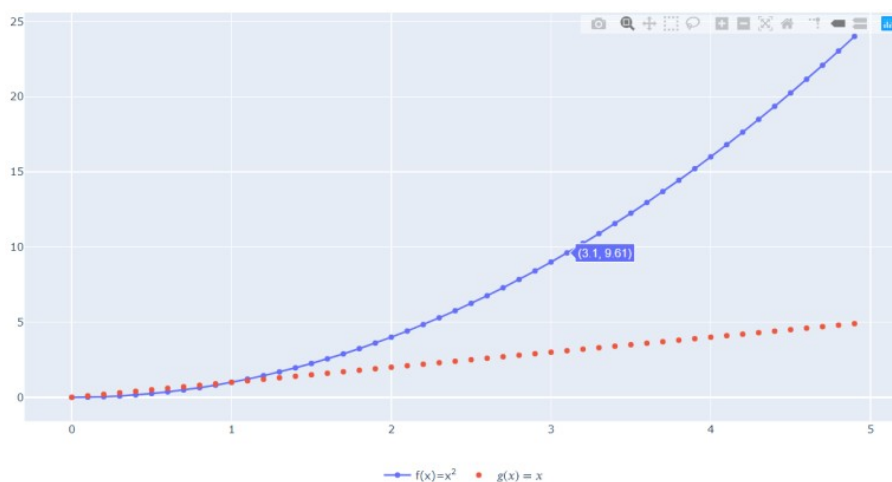


Рисунок 6 - Сложный интерактивный график
DOI: <https://doi.org/10.60797/IRJ.2026.164.70.6>

Анимации в Plotly создаются через последовательное отображение фреймов (листинг 3, рис. 7,8).

Листинг 3. Код для построения сложного линейного графика

```
import plotly.graph_objects as go

fig = go.Figure()
fig.add_trace(go.Scatter(x=[x[0]], y=[f(x)[0]], mode='lines+markers', name='f(x)=x2'))
frames = []
for i in range(1, len(x)):
    frames.append(go.Frame(data=[go.Scatter(x=x[:i+1], y=f(x[:i+1]))]))
fig.frames = frames
```

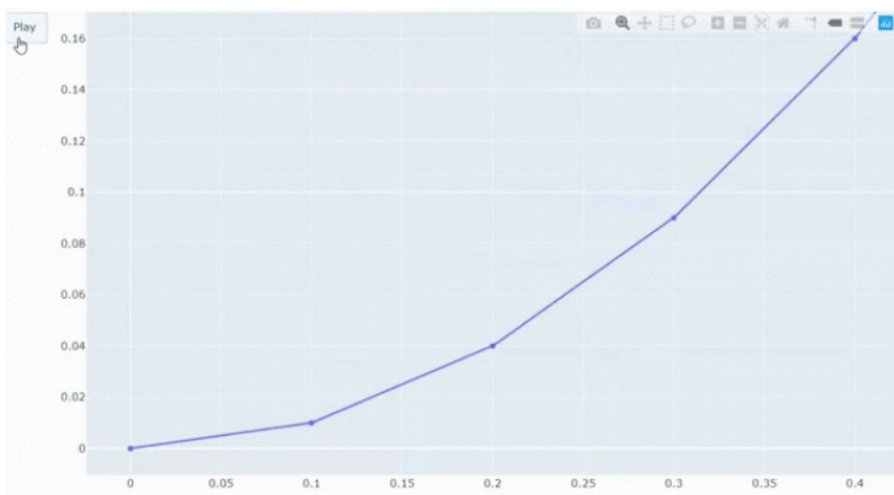


Рисунок 7 - Начальный фрейм
DOI: <https://doi.org/10.60797/IRJ.2026.164.70.7>

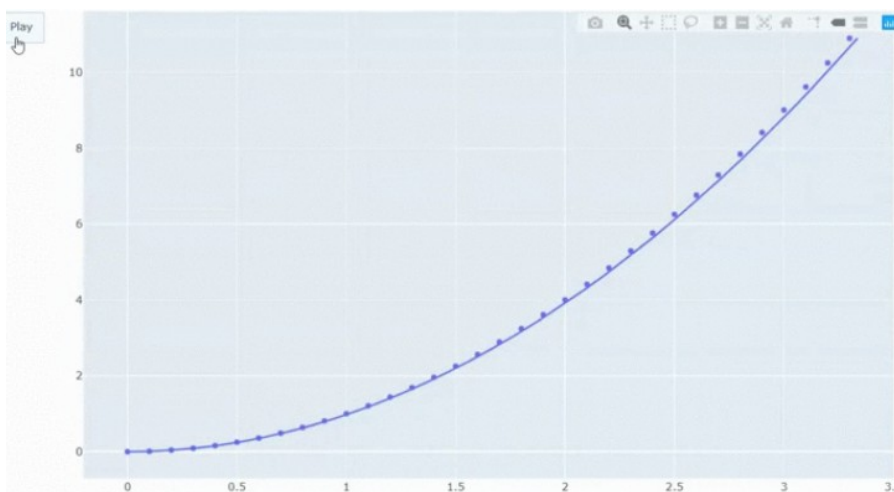


Рисунок 8 - Конечный фрейм
DOI: <https://doi.org/10.60797/IRJ.2026.164.70.8>

Matplotlib предоставляет детальный контроль над созданием статических графиков, поддерживая широкий спектр визуализаций — от базовых линейных графиков до сложных трехмерных поверхностей и специализированных диаграмм. Библиотека включает поддержку TeX-разметки, тонкую настройку осей и интеграцию с NumPy/SciPy (листинг 4) [13], [14].

Листинг 4. Код для построения графика с исп. Seaborn

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Import Data
df = pd.read_csv('https://raw.githubusercontent.com/selva86/datasets/master/mpg_ggplot.csv')
df_select = df.loc[df.cyl.isin([4, 8]), :]

# Plot
sns.set_style("white")
gridobj = sns.lmplot(x="displ", y="hwy", hue="cyl", data=df_select,
                    height=7, aspect=1.6, robust=True, palette="tab10",
                    scatter_kws=dict(s=60, linewidths=0.7, edgecolors='black'))

# Decorations
gridobj.set(xlim=(0.5, 7.5), ylim=(0, 50))
plt.title("Scatterplot with line of best fit grouped by number of cylinders", fontsize=20)
plt.show()
```


Данный подход позволяет наглядно оценить различия в характере зависимостей между анализируемыми переменными для разных групп наблюдений, выявить аномальные значения, а также сравнить углы наклона регрессионных прямых (рис. 9).

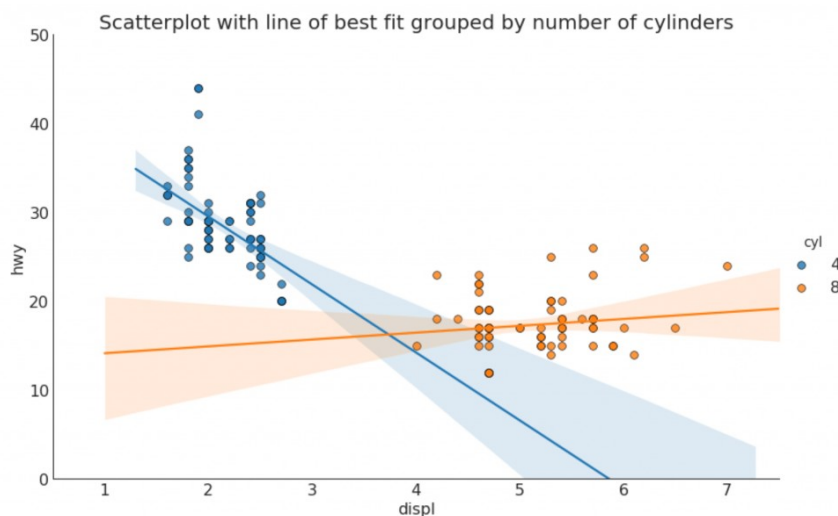


Рисунок 9 - График с использованием Seaborn
DOI: <https://doi.org/10.60797/IRJ.2026.164.70.9>

Упорядоченная гистограмма представляет собой эффективный инструмент визуализации данных, позволяющий наглядно отобразить порядок ранжирования элементов. В данном исследовании рассматривается процесс создания такой гистограммы с использованием библиотеки Matplotlib в Python, а также добавление значений показателей над диаграммой для обеспечения точной и детализированной информации (листинг 5) [15], [16].

Листинг 5. Код для подготовки данных

```
import pandas as pd

# Загрузка данных
df_raw = pd.read_csv('https://github.com/selva86/datasets/raw/master/mpg_ggplot2.csv')
df = df_raw[['cty', 'manufacturer']].groupby('manufacturer').apply(lambda x: x.mean())
df.reset_index(inplace=True)
```

После подготовки данных можно приступить к построению гистограммы с использованием Matplotlib. Создается фигура и оси, после чего добавляется гистограмма с помощью метода bar (листинг 6, 7; рис. 10, 11).

Листинг 6. Код для подготовки визуализации

```
import matplotlib.pyplot as plt
import matplotlib.patches as patches

# Создание фигуры и осей
fig, ax = plt.subplots(figsize=(16, 8), facecolor='White', dpi=80)

# Вертикальные линии
ax.vlines(x=df.index, ymin=0, ymax=df.cty, color='firebrick', alpha=0.7, linewidth=2)

# Аннотация текста
for i, cty in enumerate(df.cty):
    ax.text(i, cty+0.5, round(cty, 1), horizontalalignment='center')

# Настройка заголовка, меток и тиков
ax.set_title('Bar Chart for Highway Mileage', fontdict={'size': 22})
ax.set_ylabel('Miles Per Gallon', y=0.8)
ax.set_xticks(df.index)
ax.set_xticklabels(df.manufacturer.str.upper(), rotation=60, horizontalalignment='center')
```

Листинг 7. Добавление цветowych патчей

```
import matplotlib.pyplot as plt
import matplotlib.patches as patches

# Добавление цветowych патчей
p1 = patches.Rectangle((0.57, -0.005), width=0.33, height=0.13, alpha=0.1, facecolor='green')
p2 = patches.Rectangle((1.124, -0.005), width=0.446, height=0.1, alpha=0.1, facecolor='green')

fig.add_artist(p1)
fig.add_artist(p2)

# Отображение графика
plt.show()
```

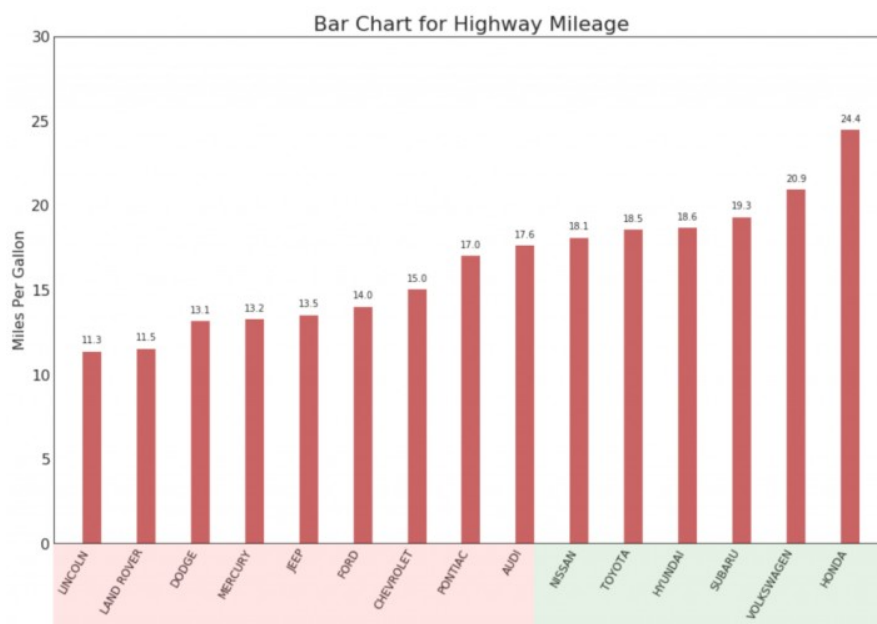


Рисунок 10 - Упорядоченная гистограмма
DOI: <https://doi.org/10.60797/IRJ.2026.164.70.10>

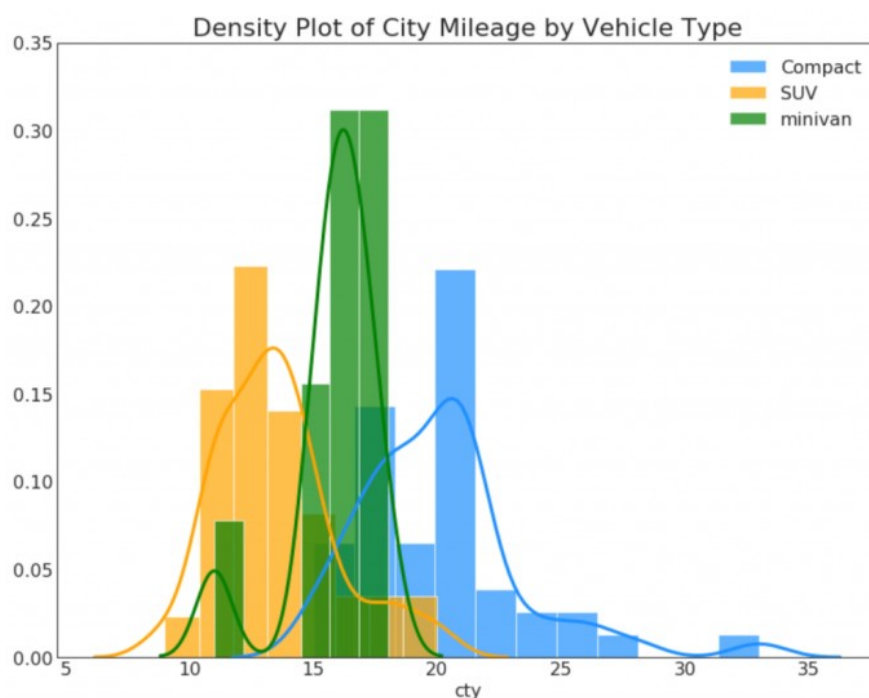


Рисунок 11 - Комбинированный график
DOI: <https://doi.org/10.60797/IRJ.2026.164.70.11>

Обе библиотеки эффективно интегрируются с Pandas, обеспечивая работу с разнообразными типами данных. Plotly ориентирована на динамическое взаимодействие, а Matplotlib предоставляет точные инструменты для научной графики, что делает их мощным комбинированным решением для анализа данных [17].

Оптимизация работы с данными и визуализация больших массивов

Оптимизация работы с данными критически важна для повышения продуктивности, особенно при анализе больших наборов данных. Масштабируемость в Data Science напрямую связана с возможностью параллельной обработки данных, что позволяет распределять нагрузку между несколькими ядрами процессора.

В языке Python для параллельных вычислений обычно используется модуль multiprocessing, однако в Jupyter Notebook его применение нередко вызывает затруднения из-за конфликтов при управлении дочерними процессами. В таких случаях удобной альтернативой является библиотека Joblib, которая обеспечивает параллельное выполнение задач за счёт фоновых процессов, работающих независимо от основного процесса Jupyter [18], [19].

Для демонстрации анализа задач с длительным временем выполнения можно использовать алгоритмы с высокой временной сложностью, такие как «monkey sort» (или «bogosort») — крайне неэффективный метод сортировки. Алгоритм проверяет, отсортирован ли массив; если нет, случайным образом перемешивает элементы до достижения отсортированного порядка.

Средняя временная сложность составляет $O((n+1)!)$ из-за вероятностной природы алгоритма, что означает факториальное количество операций для сортировки массива из n элементов. Фактическое время выполнения может варьироваться из-за случайного характера перемешивания [21], [22].

После загрузки необходимых библиотек реализуем алгоритм «monkey sort» на Python (листинг 8).

Листинг 8. Код алгоритма monkey sort

```
import random

def bogosort(arr):
    def correct(arr, comparator=lambda x: x):
        for i in range(1, len(arr)):
            if comparator(arr[i - 1]) - comparator(arr[i]) > 0:
                return False
        return True

    while not correct(arr):
        random.shuffle(arr)
    return arr
```

Для проведения тестирования гипотез и анализа данных требуется создание набора тестовых данных. В данном исследовании генерируется двумерный массив, содержащий 1000 наборов данных, каждый из которых включает 8 случайно расположенных целочисленных значений (листинг 9).

Листинг 9. Набор тестовых данных

```
import numpy as np
# Создание массива случайных чисел
bigdata = np.array([[np.random.randint(0, 100) for _ in range(1000)] for _ in range(1000)])

# Вывод первых 5 элементов
print(bigdata[:5])

array([[24, 77, 45, 37, 47, 52, 80, 95],
       [53, 40, 81, 24, 12, 20, 80, 42],
       [34, 58, 90, 41, 85, 87, 80, 41],
       [86, 17, 18, 60, 72, 89, 48, 28],
       [ 1, 38, 44, 12, 97, 46, 52, 52]])
```

Проверить работу алгоритма можно на этом наборе данных, для учета времени используем встроенную функцию Python%%time (рис. 12).

```
[Parallel(n_jobs=6)]: Using backend LokyBackend with 6 concurrent workers.
[Parallel(n_jobs=6)]: Done  2 out of  6 | elapsed:  54.9s remaining:  1.8min
[Parallel(n_jobs=6)]: Done  3 out of  6 | elapsed:  58.0s remaining:  58.0s
[Parallel(n_jobs=6)]: Done  4 out of  6 | elapsed:  59.8s remaining:  29.8s
CPU times: total: 62.5 ms
Wall time: 1min 3s
[Parallel(n_jobs=6)]: Done  6 out of  6 | elapsed:  1.1min remaining:  0.0s
[Parallel(n_jobs=6)]: Done  6 out of  6 | elapsed:  1.1min finished
```

Рисунок 12 - Проверка результатов времени сортировки без параллелизма
DOI: <https://doi.org/10.60797/IRJ.2026.164.70.12>

Сортировка завершилась за 4 минуты 32 секунды. Теперь решим ту же задачу, применив параллелизм посредством разбиения данных на партии [23].

Используется библиотека Joblib в сочетании с лямбда-функциями Python, что позволяет эффективно выполнять вычисления без изоляции процессов и последующей синхронизации с общей очередью сбора данных. Joblib автоматически разделяет данные и собирает результаты через класс Parallel, распараллеливая задачи между ядрами процессора без необходимости явного управления процессами (рис. 13).

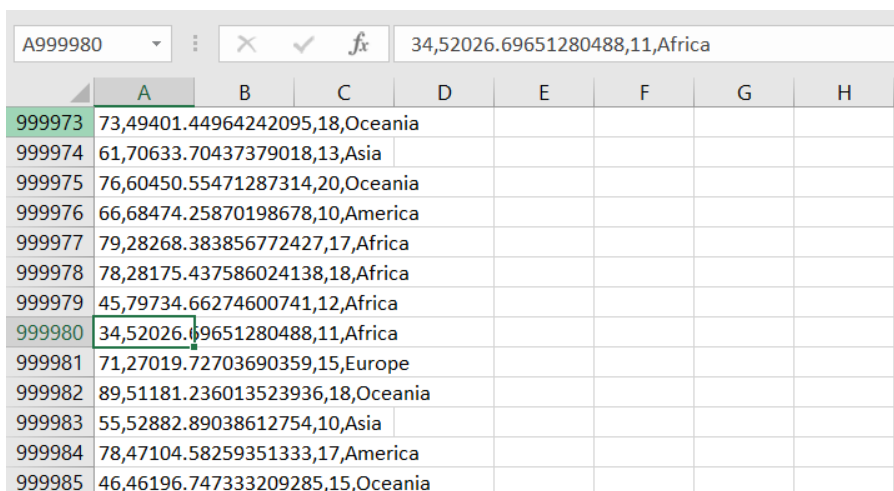
```
[Parallel(n_jobs=12)]: Using backend LokyBackend with 12 concurrent workers.
[Parallel(n_jobs=12)]: Done  1 tasks      | elapsed:  31.1s
[Parallel(n_jobs=12)]: Done  3 out of 12 | elapsed:  35.1s remaining:  1.8min
[Parallel(n_jobs=12)]: Done  5 out of 12 | elapsed:  36.1s remaining:  50.6s
[Parallel(n_jobs=12)]: Done  7 out of 12 | elapsed:  36.5s remaining:  26.0s
[Parallel(n_jobs=12)]: Done  9 out of 12 | elapsed:  38.0s remaining:  12.6s
[Parallel(n_jobs=12)]: Done 12 out of 12 | elapsed:  44.9s finished
```

Рисунок 13 - Проверка результатов времени сортировки с параллелизмом
DOI: <https://doi.org/10.60797/IRJ.2026.164.70.13>

Применение параллельных вычислений ускорило обработку в 6 раз (с 272 секунд до 44.9).

Визуализация больших данных в Jupyter Notebook

Традиционные приложения типа Excel демонстрируют ограничения при работе с большими данными (>600000 записей): экспоненциальный рост времени обработки, частые сбои и невозможность построения сложных визуализаций (рис. 14).



	A	B	C	D	E	F	G	H
999973	73,49401.44964242095,18,Oceania							
999974	61,70633.70437379018,13,Asia							
999975	76,60450.55471287314,20,Oceania							
999976	66,68474.25870198678,10,America							
999977	79,28268.383856772427,17,Africa							
999978	78,28175.437586024138,18,Africa							
999979	45,79734.66274600741,12,Africa							
999980	34,52026.69651280488,11,Africa							
999981	71,27019.72703690359,15,Europe							
999982	89,51181.236013523936,18,Oceania							
999983	55,52882.89038612754,10,Asia							
999984	78,47104.58259351333,17,America							
999985	46,46196.747333209285,15,Oceania							

Рисунок 14 - Фрагмент файла Excel
DOI: <https://doi.org/10.60797/IRJ.2026.164.70.14>

При работе с крупными массивами данных Excel демонстрирует значительные ограничения: операции замедляются, приложение зависает или перестает отвечать. Это особенно проявляется при построении графиков или фильтрации сотен тысяч/миллионов строк, поскольку Excel не рассчитан на обработку таких объемов данных.

Кроме того, Excel может выводить неполную информацию без ошибок, что ограничивает его использование для анализа больших данных (рис. 15).



```
[12]: # Загрузка файла citizens.csv
csv_path = 'citizens.csv' # сюда загрузи файл citizens.csv через "Файлы" @ Colab
df = pd.read_csv(csv_path)

print(f"Количество строк: {len(df)}")
df.head()
```

Количество строк: 1000000

	age	income	education_years	region
0	69	55461.809444	13	Europe
1	32	66953.234167	17	Africa
2	89	81496.710485	10	Africa
3	78	33337.379581	13	Asia
4	38	44647.701678	9	America

Рисунок 15 - Вывод неполной информации с файла Excel
DOI: <https://doi.org/10.60797/IRJ.2026.164.70.15>

Связка Jupyter Notebook с базами данных (PostgreSQL/MySQL) решает эти проблемы. Далее нужно загрузить данные в PostgreSQL (листинг 10,11).

Листинг 10. Подключение к PostgreSQL в Jupyter notebook

```
import numpy as np

# Подключение к PostgreSQL из Python

[:]: !pip install psycopg2-binary
from sqlalchemy import create_engine
import pandas as pd

# Создаем подключение
engine = create_engine('postgresql://postgres:password@localhost:5432/citizen_db')
```

Листинг 11. Подключение к PostgreSQL в Jupyter notebook.

```
#Загрузка данных в PostgreSQL
[:]: # Загрузка данных в базу данных (таблица citizens)
df.to_sql('citizens', engine, if_exists='replace', index=False)
```

В результате обработки данных удалось успешно построить график распределения возрастов, включающий все 1 миллион элементов. Процесс обработки и создания графика (рис. 16) занял всего одну минуту реального времени, что демонстрирует эффективность используемых методов и инструментов для работы с большими объемами данных.

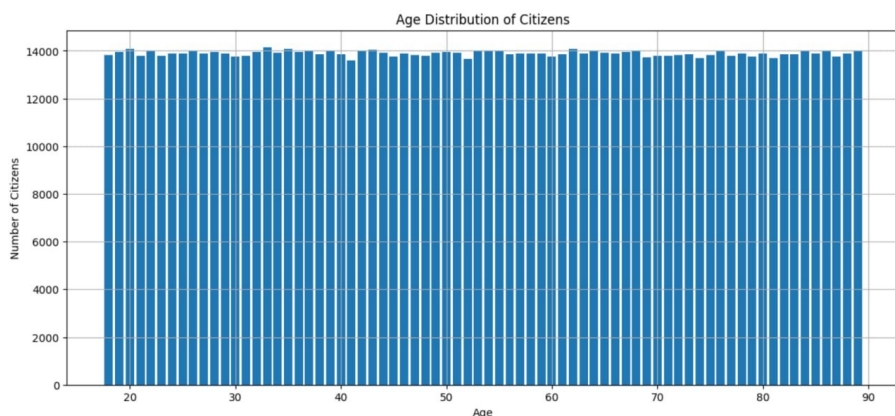


Рисунок 16 - График распределения возрастов
DOI: <https://doi.org/10.60797/IRJ.2026.164.70.16>

В рамках данного исследования также была успешно реализована визуализация данных с использованием трехмерного графика, отображающего взаимосвязь между возрастом, уровнем образования и средним доходом. Для этого был использован язык запросов SQL для извлечения необходимых данных из базы данных, а также библиотека Plotly для создания интерактивного 3D-графика.

С помощью SQL-запроса были извлечены данные о возрасте, уровне образования и среднем доходе из таблицы citizens. Запрос включал группировку данных по возрасту и уровню образования, а также вычисление среднего дохода для каждой группы (листинг 12).

Листинг 12. Извлечение данных

```
import plotly.graph_objects as go
query = """
SELECT age, education_years, AVG(income) as avg_income
FROM citizens
GROUP BY age, education_years
"""
df_3d = pd.read_sql(query, engine)
```

Для визуализации данных был использован модуль plotly.graph_objects, который позволяет создавать интерактивные графики. В частности, был создан 3D-график рассеяния, где по осям X, Y и Z отображаются возраст, уровень образования и средний доход соответственно (листинг 13).

Листинг 13. Код для построения 3D графика

```
df_3d = pd.read_sql(query, engine)

fig = go.Figure(data=[go.Scatter3d(
    x=df_3d['age'],
    y=df_3d['education_years'],
    z=df_3d['avg_income'],
    mode='markers',
    marker=dict(size=2)
)])

fig.update_layout(scene=dict(
    xaxis_title='Age',
    yaxis_title='Education Years',
    zaxis_title='Average Income',
    margin=dict(l=0, r=0, b=0, t=0))
```

```
fig.show()
```

3D-график, отображающий распределение данных о возрасте, уровне образования и среднем доходе. График позволяет наглядно оценить взаимосвязь между этими параметрами и выявить возможные закономерности и тенденции. Использование интерактивных возможностей Plotly позволяет исследователям динамически взаимодействовать с графиком, изменяя угол обзора и масштаб, что значительно облегчает процесс анализа и интерпретации данных (рис. 17).

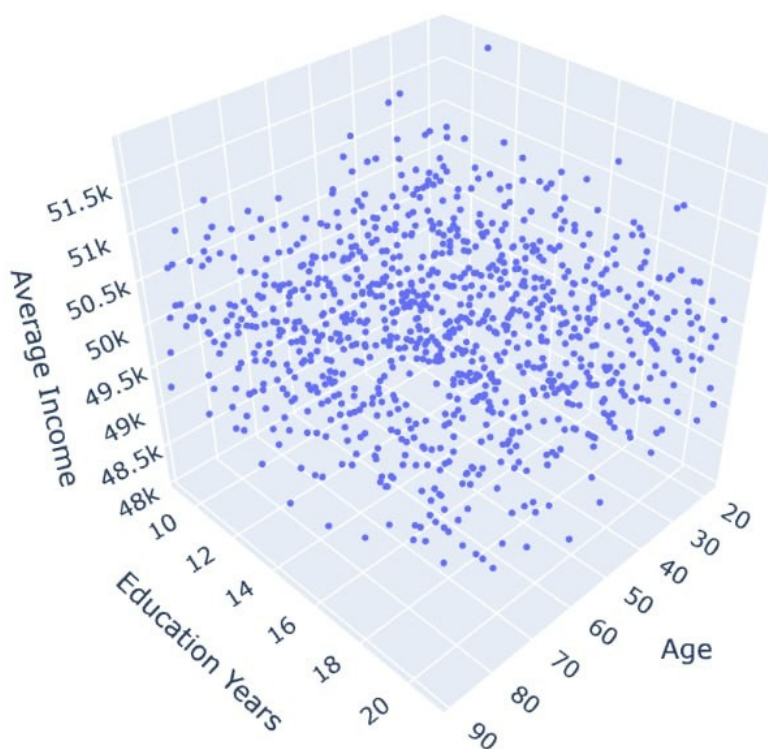


Рисунок 17 - 3D-график: возраст, образование, средний доход
DOI: <https://doi.org/10.60797/IRJ.2026.164.70.17>

Архитектурная оптимизация (сервер БД выполняет ресурсоемкие операции, Jupyter обеспечивает визуализацию) позволяет эффективно работать с выборками >1 млн. записей.

Автоматизация отчетов и практическое применение методики

Jupyter Notebook поддерживает развитую систему экспорта результатов в различные форматы, что обеспечивает удобное и профессиональное представление данных для разных аудиторий. Инструмент nbconvert предоставляет широкие возможности преобразования файлов .ipynb в разнообразные выходные форматы, адаптированные под конкретные потребности и сценарии использования.

В академической и научной среде Jupyter Notebook зарекомендовал себя как незаменимый инструмент для подготовки воспроизводимых публикаций. Поддержка экспорта в форматы PDF и LaTeX позволяет исследователям создавать материалы, полностью соответствующие строгим требованиям рецензируемых научных журналов. Особую ценность представляет сохранение математических формул в нотации LaTeX, что обеспечивает корректное и профессиональное оформление сложных теоретических выкладок и математических моделей.

Для образовательных целей особенно востребован экспорт в HTML, который позволяет формировать интерактивные учебные материалы с сохранением визуализаций и возможностью демонстрации вычислительных процессов. При подготовке докладов и представлении результатов на научных конференциях формат Reveal.js открывает широкие возможности для создания динамичных веб-презентаций, объединяющих программный код, графические иллюстрации и пояснительный текст в едином интерактивном пространстве.

Для документирования аналитических моделей и методик особенно полезен экспорт в формат Markdown, который упрощает интеграцию с системами управления знаниями и обеспечивает возможность версионного контроля посредством платформ наподобие Git. В бизнес-среде востребован экспорт визуализаций в PNG и PDF форматы, что позволяет создавать высококачественные графики и диаграммы для отчетов и презентаций (рис. 18,19).

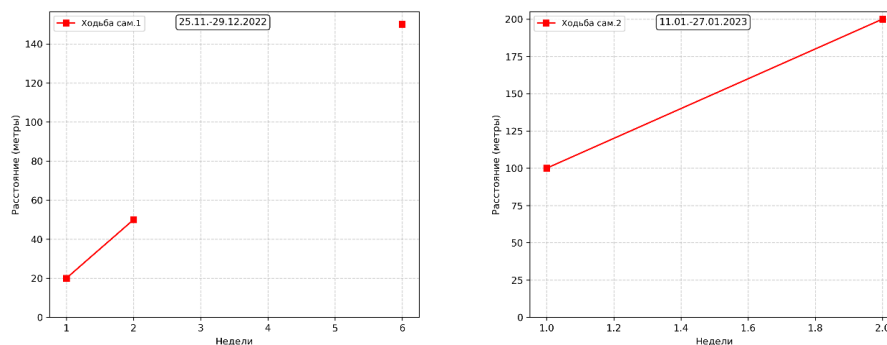


Рисунок 18 - Пример экспорта в формате png
DOI: <https://doi.org/10.60797/IRJ.2026.164.70.18>

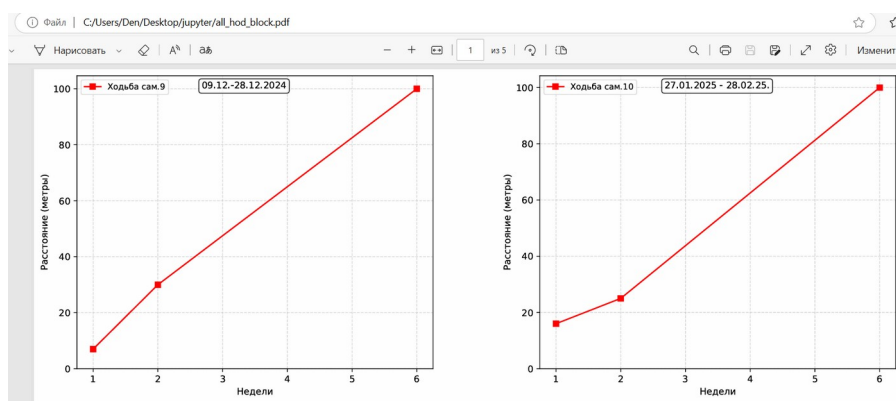


Рисунок 19 - Пример экспорта в формате pdf
DOI: <https://doi.org/10.60797/IRJ.2026.164.70.19>

Практическое применение разработанной методики было продемонстрировано на реальной задаче визуализации данных с учетом объединенных статистических параметров. Данные включали такие показатели, как среднее арифметическое, стандартное отклонение и количество выделяемой памяти в куче за один вызов, а также требовали построения графика зависимости от числа рекурсивных вызовов.

Для реализации задачи данные были преобразованы из исходного формата в Excel таблицу, а затем в CSV формат для последующей загрузки в Jupyter Notebook. Процесс обработки включал переименование столбцов для удобства работы, заполнение пропущенных значений и создание дополнительного параметра «сложность рекурсии» как суммы параметров m и n . Успешная загрузка данных подтвердила готовность к последующему анализу и визуализации.

Визуализация была реализована с использованием библиотеки Matplotlib, что позволило создать всеобъемлющий график, отображающий три ключевых параметра одновременно. По оси X отображался номер измерения, по оси Y — среднее время выполнения метода в наносекундах. Размер точек визуализировал объем выделенной памяти при конкретном вызове метода, вертикальные линии показывали стандартное отклонение, а цвет точек отражал суммарную «сложность рекурсии». Такое многомерное представление данных позволило наглядно продемонстрировать зависимость времени выполнения метода от числа рекурсивных вызовов и оценить влияние различных параметров на производительность и использование памяти.

Практическое применение: визуализация данных в Jupyter Notebook

Jupyter Notebook предоставляет широкие возможности для визуализации данных, включая работу с табличной информацией из различных источников. На рисунке 20 показан пример типовых данных, импортированных в рабочую среду.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	X	Y	Москва	Питер	Казань	Челобоксары	Новосибирск	Екатеринбург	Красноярск	Нижегород	Челябинск	Уфа	Самара
2	2009	100 000	150 077	94 974	57 753	30 213	52 477	47 909	46 088	47 860	48 100	47 095	50 597
3	2010	150 000	121 415	84 451	40 577	31 932	47 456	49 218	38 674	47 370	37 591	46 723	40 508
4	2011	200 000	135 887	82 689	43 402	35 032	48 144	52 508	44 246	43 466	35 373	46 139	45 018
5	2012	250 000	178 065	82 731	51 325	43 531	49 789	59 158	51 553	47 444	38 244	53 551	48 775
6	2013	300 000	205 523	94 745	57 311	49 120	52 490	69 131	57 244	61 210	43 145	61 547	54 724
7	2014	350 000	220 937	98 825	62 479	50 571	59 404	72 656	62 455	67 977	46 099	69 209	60 908
8	2015	400 000	235 445	102 826	65 026	49 371	60 960	75 240	58 438	70 989	45 226	70 135	64 443
9	2016	450 000	215 833	102 461	64 645	45 778	55 365	69 300	54 978	67 308	42 193	63 465	63 452
10	2017	500 000	207 473	106 471	65 276	43 883	55 780	67 173	49 010	64 007	39 940	59 082	57 860
11	2018	550 000	222 949	114 652	69 723	42 848	56 079	70 745	51 013	64 378	38 951	58 881	59 116
12	2019	600 000	239 405	116 519	77 362	42 004	69 438	72 613	57 158	67 922	40 686	66 109	65 360
13	2020	650 000	264 102	120 639	85 492	43 097	67 147	73 097	64 087	73 143	39 833	70 913	61 490
14	2021	700 000	311 102	163 474	110 420	55 785	81 925	86 551	78 508	95 478	45 653	77 882	80 226
15	2022	750 000	386 902	203 677	138 665	75 223	110 722	102 384	96 323	118 011	58 248	87 983	117 163
16	2023	800 000	406 515	231 305	159 456	96 175	120 963	107 609	110 067	131 029	70 195	100 616	131 718
17	2024	850 000	424 238	235 277	162 983	95 505	130 796	120 807	123 211	148 270	88 314	122 044	127 111
18	2025	900 000	429 725	234 534	201 191	96 208	125 709	90 111	121 998	148 584	86 101	141 013	138 039

Рисунок 20 - Данные в Excel в привычном виде

DOI: <https://doi.org/10.60797/IRJ.2026.164.70.20>

Для создания визуализаций используется библиотека Plotly, которая позволяет легко создавать интерактивные графики. Пример создания круговой диаграммы представлен в листинге 14.

Листинг 14. Код для создания круговой диаграммы

```
fig = go.Figure()
fig.add_trace(go.Pie(values=sum_counts, labels=sum_counts.index, sort=False))
fig.show()
```

Этот фрагмент демонстрирует, насколько легко и удобно можно создавать и работать с круговыми диаграммами в Jupyter Notebook (рис. 21).

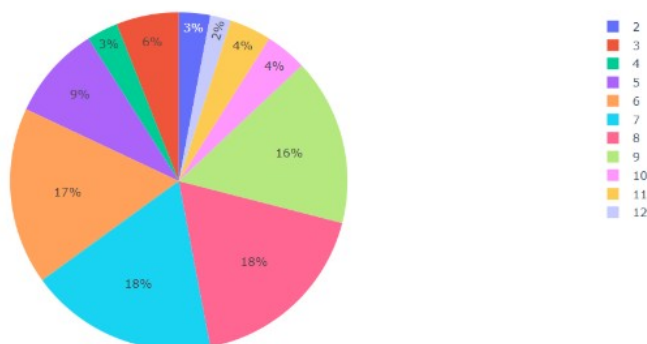


Рисунок 21 - Простая круговая диаграмма

DOI: <https://doi.org/10.60797/IRJ.2026.164.70.21>

Для выделения отдельных секторов используется параметр pull (см. листинг 15, рис. 22).

Листинг 15. Код для создания круговой диаграммы с выделенным сектором

```
fig = go.Figure()
pull = [0]*len(sum_counts)
pull[sum_counts.tolist().index(sum_counts.max())] = 0.2
fig.add_trace(go.Pie(values=sum_counts, labels=sum_counts.index, pull=pull))
fig.show()
```

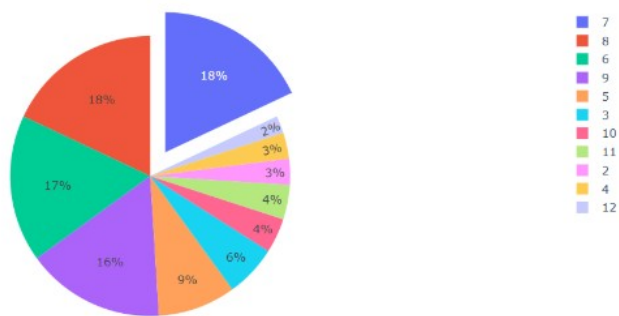



Рисунок 22 - Круговая диаграмма с выделенным сектором
DOI: <https://doi.org/10.60797/IRJ.2026.164.70.22>

Преобразование в кольцевую диаграмму достигается параметром `hole` (см. листинг 16, рис. 23).
Листинг 16. Код для создания круговой диаграммы без центральной части

```
fig = go.Figure()
pull = [0]*len(sum_counts)
pull[sum_counts.tolist().index(sum_counts.max())] = 0.2
fig.add_trace(go.Pie(values=sum_counts, labels=sum_counts.index, pull=pull, hole=0.9))
fig.show()
```

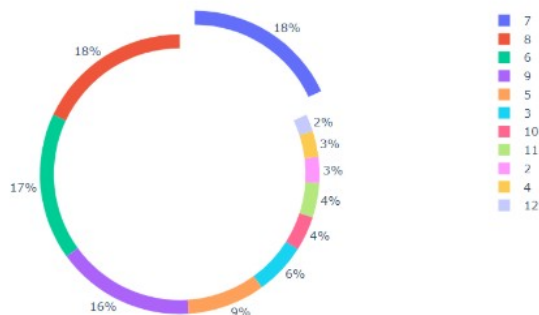


Рисунок 23 - Круговая диаграмма без центральной части
DOI: <https://doi.org/10.60797/IRJ.2026.164.70.23>

Для работы с реальными данными используется связка библиотек `pandas` и `matplotlib`. После загрузки и предобработки формируется комплексный график, отображающий несколько параметров одновременно (см. листинг 17, рис. 24).

Листинг 17. Код для загрузки данных в Jupyter notebook

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

sns.set(style="whitegrid")

# Загрузка CSV напрямую
df = pd.read_csv(r'data2copy.csv')
```

```
# Переименуем при необходимости
rename_map = {
    "RecCalls": "Recurse_calls",
    "AllocMem_bytes": "Memory_allocated"
}
df.rename(columns=rename_map, inplace=True)

# Заполняем пропуски
df["Memory_allocated"] = df["Memory_allocated"].fillna(0)
df["rec_complexity"] = df["m"] + df["n"]

df.head()
```

[2]:

	Method	Recurse_calls	m	n	Mean_ns	StdDev_ns	Memory_allocated	rec_complexity
0	Baseline	4	1	1	3.506	0.0902	0.0	2
1	ValueTask	4	1	1	76.036	2.2531	0.0	2
2	IValueTaskSource	4	1	1	378.316	11.3795	368.0	2
3	Task	4	1	1	50.700	1.0317	0.0	2
4	Baseline	6	1	2	7.313	0.0782	0.0	3

Рисунок 24 - Положительный результат выполнения кода
DOI: <https://doi.org/10.60797/IRJ.2026.164.70.24>

На данном этапе мы используем библиотеку Matplotlib для создания графика, который позволяет визуализировать ключевые статистические параметры (см. листинг 18, рис. 25).

Листинг 18. Код для создания основного графика

```
plt.figure(figsize=(12, 8))
scatter = plt.scatter(
    df.index,
    df["Mean_ns"],
    s=df["Memory_allocated"] / 100,
    c=df["rec_complexity"],
    cmap="plasma",
    alpha=0.7
)
for i in df.index:
    plt.errorbar(
        x=i,
        y=df.loc[i, "Mean_ns"],
        yerr=df.loc[i, "StdDev_ns"],
        fmt='none',
        ecolor='black',
        capsize=3
    )

plt.colorbar(scatter, label='Число рекурсивных вызовов')
plt.xlabel("Номер измерения")
plt.ylabel("Среднее время (нс)")
plt.title("Зависимость времени от числа рекурсивных вызовов, отклонение, память, рекурсия")
plt.tight_layout()
plt.show()
```

В первом блоке создается точечная диаграмма с возможностью настраивать размер точек, их цвет, прозрачность. Второй блок добавляет вертикальные линии внутри точек (кругов). Завершающая часть кода реализует оформление, легенду графика, она включает в себя подписи осей и цветовое оформление. Функция `plt.colorbar()` создает цветную шкалу, которая визуальнo связывает градиентную палитру маркеров с соответствующими значениями сложности рекурсивных вычислений.

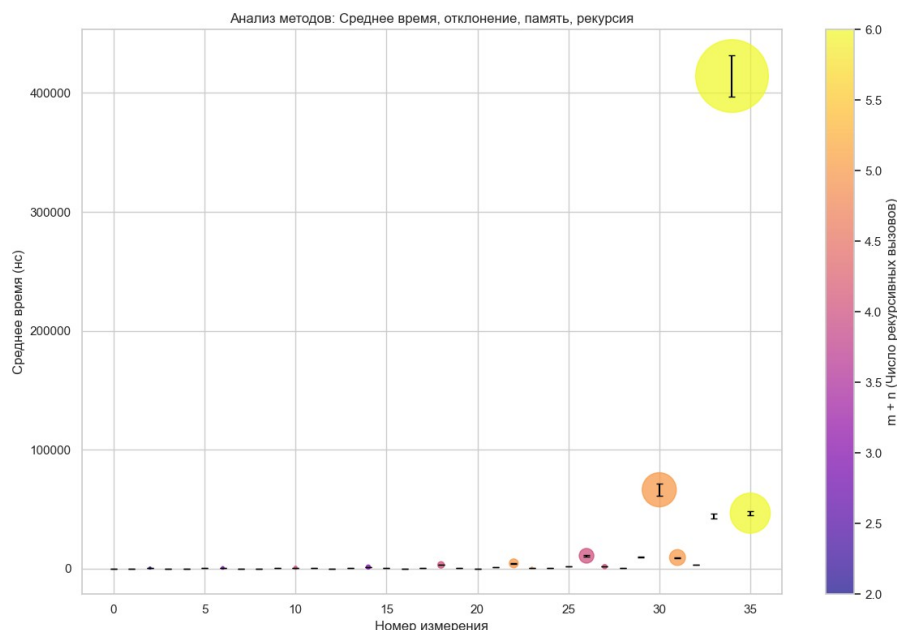


Рисунок 25 - Результат выполнения кода
DOI: <https://doi.org/10.60797/IRJ.2026.164.70.25>

График позволяет одновременно анализировать:

- 1) среднее время выполнения (ось Y);
- 2) объем используемой памяти (размер точек);
- 3) сложность вычислений (цвет точек);
- 4) стандартное отклонение (вертикальные линии).

Такой подход демонстрирует преимущества Jupyter Notebook для работы со сложными многомерными данными по сравнению с традиционными инструментами.

Заключение

В результате проведенного исследования было показано, что Jupyter Notebook представляет собой универсальную и эффективную среду для визуализации статистических данных, позволяющую объединять вычислительные возможности Python с гибкими инструментами графического представления информации. Рассмотренные библиотеки Plotly и Matplotlib продемонстрировали высокую результативность как для статической научной графики, так и для создания интерактивных решений, что делает их совместное использование оптимальным для анализа данных различной сложности и объема. Особое внимание было уделено вопросам оптимизации работы с большими массивами, применению параллельных вычислений и интеграции с системами управления базами данных, что обеспечивает масштабируемость и надежность аналитических процессов.

Практическая значимость предложенного методологического подхода проявляется в его применимости для широкого спектра задач — от образовательных и исследовательских до бизнес-аналитики и промышленных проектов. Экспорт визуализаций в разнообразные форматы, включая HTML, PDF, LaTeX и интерактивные презентации, открывает возможности для подготовки воспроизводимых публикаций, учебных материалов и профессиональных отчетов. Важным результатом является формирование принципов построения визуализаций, учитывающих когнитивные особенности восприятия и направленных на повышение точности интерпретации данных и качества принимаемых решений.

Перспективы дальнейшего развития работы связаны с расширением функционала Jupyter Notebook путем интеграции методов машинного обучения и искусственного интеллекта для автоматического выбора оптимальных способов визуализации в зависимости от структуры и характеристик данных. Представляется актуальным внедрение интеллектуальных рекомендательных систем, способных адаптировать цветовые схемы, масштабируемые графики и интерактивные элементы под конкретные задачи пользователя. Дополнительным направлением может стать развитие облачных решений и коллаборативных сервисов на основе Jupyter, что позволит организовать совместную работу исследователей и аналитиков в реальном времени. В долгосрочной перспективе внедрение описанного подхода способно повысить эффективность анализа данных, сделать визуализацию более доступной и интуитивно понятной, а также обеспечить новые возможности для научных исследований и прикладных разработок.

**Конфликт интересов**

Не указан.

Рецензия

Сообщество рецензентов Международного научно-исследовательского журнала
DOI: <https://doi.org/10.60797/IRJ.2026.164.70.26>

Conflict of Interest

None declared.

Review

International Research Journal Reviewers Community
DOI: <https://doi.org/10.60797/IRJ.2026.164.70.26>

Список литературы / References

1. Карелин И.В. Выбор оптимального варианта визуализации данных / И.В. Карелин, С.И. Дмитриенко, Е.С. Грошева // Профессиональная ориентация. — 2022. — № 1. — С. 12–16.
2. Makarova E.A. Approaches to visualizing big text data at the stage of collection and pre-processing / E.A. Makarova, D.G. Lagerev, F.Y. Lozbinnev // Scientific Visualization. — 2019. — Vol. 11. — № 4. — P. 13–26. — DOI: 10.26583/sv.11.4.02.
3. Екубджонов Д.И. Исследование производительности технологий объектно-реляционного отображения при взаимодействии с Microsoft SQL Server / Д.И. Екубджонов, Р.Ф. Гибадуллин // Международный научно-исследовательский журнал. — 2024. — № 10 (148). — DOI: 10.60797/IRJ.2024.148.145.
4. Agapova O.Yu. Analysis of the clinical data in the jupyter environment / O.Yu. Agapova, K.M. Laufer, A.D. Zuev [et al.] // 11th IEEE International Conference on Application of Information and Communication Technologies, AICT 2017. — Moscow, 2019. — P. 8686840. — DOI: 10.1109/ICAICT.2017.8686840.
5. Vaslavskaya I. The Use of Blockchain Technology for Transport and Logistics Systems in the Digital Economy / I. Vaslavskaya, I. Koshkina, R. Zaripova // Finance, Economics, and Industry for Sustainable Development. — Springer, Cham. 2024. — DOI: 10.1007/978-3-031-56380-5_16.
6. Ханов М.Н. Использование визуализации данных для анализа статистической информации / М.Н. Ханов, Н.Г. Яковлева, Н.В. Попов // Информационные ресурсы и системы в экономике, науке и образовании : сборник статей XIV Международной научно-практической конференции, Пенза, 29–30 апреля 2024 года. — Пенза: Приволжский Дом знаний, 2024. — С. 188–192.
7. Шигабетдинова Д.И. Обнаружение и локализация утечек на нефтедобывающих объектах с помощью компьютерного зрения / Д.И. Шигабетдинова, З.М. Гизатуллин, М.П. Шлеймович // Вестник Технологического университета. — 2025. — Т. 28. — № 5. — С. 123–128. — DOI: 10.55421/3034-4689_2025_28_5_123.
8. Давыдов И.П. Библиотеки интерактивной визуализации данных в Python / И.П. Давыдов, В.Л. Лукинов // Научное и техническое творчество молодежи : Материалы Всероссийской научно-практической конференции с международным участием, Новосибирск, 19–20 апреля 2023 года / Под ред. А.В. Ефимова, Т.И. Монастырской. — Новосибирск: Сибирский государственный университет телекоммуникаций и информатики, 2023. — Ч. 2. — С. 95–99.
9. Шарипов Р.Р. Разработка программного комплекса потокового шифра RC4 для обучающихся по дисциплине «криптография» / Р.Р. Шарипов, С.П. Макаров, А.А. Кассирова // Международный научно-исследовательский журнал. — 2024. — № 9 (147). — DOI: 10.60797/IRJ.2024.147.15.
10. Лежебоков А.А. Технологии визуализации для прикладных задач интеллектуального анализа данных / А.А. Лежебоков, Э.В. Кулиев // Известия Кабардино-Балкарского научного центра РАН. — 2019. — № 4 (90). — С. 14–23. — DOI: 10.35330/1991-6639-2019-4-90-14-23.
11. Шкиндеров М.С. Моделирование помехоустойчивости системы контроля и управления доступом при воздействии электростатического разряда / М.С. Шкиндеров, Р.Р. Мубараков // Труды МАИ. — 2021. — № 120. — DOI: 10.34759/trd-2021-120-12.
12. Титов А.Н. Визуализация данных в Python. Основы работы с интерактивной библиотекой Altair : Учебно-методическое пособие / А.Н. Титов, Р.Ф. Тазиева. — Казань: Казанский национальный исследовательский технологический университет, 2024. — 128 с.
13. Шурдилов И.С. Разработка системы распознавания эмоций по лицевым выражениям на основе машинного обучения / И.С. Шурдилов, М.Г. Нуриев, М.Г. Лаптева [и др.] // Международный научно-исследовательский журнал. — 2025. — № 6 (156). — DOI: 10.60797/IRJ.2025.156.52.
14. Лысенкова А.М. Современные технологии визуализации управленческих решений / А.М. Лысенкова // Актуальные аспекты модернизации российской экономики: X Всероссийская заочная научно-практическая конференция студентов, аспирантов и молодых ученых, Санкт-Петербург, 25 декабря 2023 года. — Санкт-Петербург: ЛЭТИ, 2023. — С. 598–601.
15. Катасёв А.С. Нейронечеткая модель и программный комплекс автоматизации формирования нечетких правил для оценки состояния объектов / А.С. Катасёв // Автоматизация процессов управления. — 2019. — № 1 (55). — С. 21–29.
16. Тыринов В.С. Обзор инструментов для визуализации данных / В.С. Тыринов // Молодой исследователь: от идеи к проекту: Материалы VII студенческой научно-практической конференции, Йошкар-Ола, 24–28 апреля 2023 года / Отв. ред. Д.А. Михеева. — Йошкар-Ола: Марийский государственный университет, 2023. — С. 138–143.
17. Шакирзянов Р.М. Метод автоматического позиционирования беспилотных аппаратов на основе распознавания сигнальных радиально-симметричных маркеров подводных целей / Р.М. Шакирзянов, М.П. Шлеймович, С.В. Новикова // Автоматика и телемеханика. — 2023. — № 7. — С. 93–120. — DOI: 10.31857/S0005231023070061.
18. Корнуков Н.О. Обзор возможностей библиотеки D3 для визуализации данных / Н.О. Корнуков // Молодежный научно-технический вестник. — 2015. — № 8. — С. 12.



19. Смирнов Ю.Н. Математическая модель оптимизации деятельности для цифровой системы управления предприятием / Ю.Н. Смирнов, А.В. Каляшина // Научно-технический вестник Поволжья. — 2023. — № 11. — С. 119–122.
20. Гибадуллин Р.Ф. Оптимизация асинхронных операций в .NET / Р.Ф. Гибадуллин, Д.А. Гашигуллин // Международный научно-исследовательский журнал. — 2025. — № 7 (157). — DOI: 10.60797/IRJ.2025.157.40.
21. Хабибуллин Ф.Ф. Анализ параметров ошибки положения, перемещения идеального и реального механизма для робототехнических систем / Ф.Ф. Хабибуллин, Р.Т. Исламов, Л.Ф. Хабибуллина // Проблемы машиностроения и автоматизации. — 2024. — № 1. — С. 36–43.
22. Гнатенко Ю.А. Новые технологии и инструменты для визуализации сложных статистических данных / Ю.А. Гнатенко, А.Н. Додаров // Тенденции развития науки и образования. — 2024. — № 110–17. — С. 76–83. — DOI: 10.18411/tmio-06-2024-932.
23. Гужвенко Е.И. Использование комплекса практических заданий для визуализации данных при изучении военными дисциплины «Информатика и информационные технологии в профессиональной деятельности» / Е.И. Гужвенко, В.Ю. Гужвенко // Известия Дагестанского государственного педагогического университета. Психолого-педагогические науки. — 2024. — Т. 18. — № 2. — С. 37–41. — DOI: 10.31161/1995-0659-2024-18-2-37-41.

Список литературы на английском языке / References in English

1. Karelin I.V. Vibor optimalnogo varianta vizualizatsii danih [Choosing the optimal data visualisation option] / I.V. Karelin, S.I. Dmitrienko, Ye.S. Grosheva // Professionalnaya orientatsiya [Professional Orientation]. — 2022. — № 1. — P. 12–16. [in Russian]
2. Makarova E.A. Approaches to visualizing big text data at the stage of collection and pre-processing / E.A. Makarova, D.G. Lagerev, F.Y. Lozbinev // Scientific Visualization. — 2019. — Vol. 11. — № 4. — P. 13–26. — DOI: 10.26583/sv.11.4.02.
3. Yekubdzhonov D.I. Issledovanie proizvoditel'nosti tekhnologii obektno-relyatsionnogo otobrazheniya pri vzaimodeistvii s Microsoft SQL Server [Study of the performance of object-relational mapping technologies when interacting with Microsoft SQL Server] / D.I. Yekubdzhonov, R.F. Gibadullin // Mezhdunarodnyi nauchno-issledovatel'skii zhurnal [International Research Journal]. — 2024. — № 10 (148). — DOI: 10.60797/IRJ.2024.148.145. [in Russian]
4. Agapova O.Yu. Analysis of the clinical data in the jupyter environment / O.Yu. Agapova, K.M. Laufer, A.D. Zuev [et al.] // 11th IEEE International Conference on Application of Information and Communication Technologies, AICT 2017. — Moscow, 2019. — P. 8686840. — DOI: 10.1109/ICAICT.2017.8686840.
5. Vaslavskaya I. The Use of Blockchain Technology for Transport and Logistics Systems in the Digital Economy / I. Vaslavskaya, I. Koshkina, R. Zaripova // Finance, Economics, and Industry for Sustainable Development. — Springer, Cham. 2024. — DOI: 10.1007/978-3-031-56380-5_16.
6. Khanov M.N. Ispolzovanie vizualizatsii danih dlya analiza statisticheskoi informatsii [The use of data visualisation for statistical information analysis] / M.N. Khanov, N.G. Yakovleva, N.V. Popov // Informatsionnye resursy i sistemi v ekonomike, nauke i obrazovanii : sbornik statei XIV Mezhdunarodnoi nauchno-prakticheskoi konferentsii, Penza, 29–30 aprelya 2024 goda [Information resources and systems in economics, science and education: collection of articles from the XIV International Scientific and Practical Conference, Penza, 29–30 April 2024]. — Penza: Volga House of Knowledge, 2024. — P. 188–192. [in Russian]
7. Shigabetdinova D.I. Obnaruzhenie i lokalizatsiya utechek na nefteobivayushchikh obektakh s pomoshchyu kompyuternogo zreniya [Detection and localisation of leaks at oil production facilities using computer vision] / D.I. Shigabetdinova, Z.M. Gizatullin, M.P. Shleimovich // Vestnik Tekhnologicheskogo universiteta [Bulletin of the Technological University]. — 2025. — Vol. 28. — № 5. — P. 123–128. — DOI: 10.55421/3034-4689_2025_28_5_123. [in Russian]
8. Davidov I.P. Biblioteki interaktivnoi vizualizatsii danih v Python [Libraries for interactive data visualisation in Python] / I.P. Davidov, V.L. Lukinov // Nauchnoe i tekhnicheskoe tvorchestvo molodezhi : Materiali Vserossiiskoi nauchno-prakticheskoi konferentsii s mezhdunarodnim uchastiem, Novosibirsk, 19–20 aprelya 2023 goda [Scientific and technical creativity of young people: Proceedings of the All-Russian Scientific and Practical Conference with international participation, Novosibirsk, 19–20 April 2023] / Ed. by A.V. Yefimov, T.I. Monastirskaya — Novosibirsk: Siberian State University of Telecommunications and Informatics, 2023. — Pt. 2. — P. 95–99. [in Russian]
9. Sharipov R.R. Razrabotka programmnogo kompleksa potokovogo shifra RC4 dlya obuchayushchikhsya po distsipline «kriptografiya» [Development of the RC4 stream cipher software package for students studying cryptography] / R.R. Sharipov, S.P. Makarov, A.A. Kassirova // Mezhdunarodnyi nauchno-issledovatel'skii zhurnal [International Research Journal]. — 2024. — № 9 (147). — DOI: 10.60797/IRJ.2024.147.15. [in Russian]
10. Lezhebokov A.A. Tekhnologii vizualizatsii dlya prikladnykh zadach intellektual'nogo analiza danih [Visualisation technologies for applied tasks of intelligent data analysis] / A.A. Lezhebokov, E.V. Kuliev // Izvestiya Kabardino-Balkarskogo nauchnogo tsentra RAN [Proceedings of the Kabardino-Balkarian Scientific Centre of the Russian Academy of Sciences]. — 2019. — № 4 (90). — P. 14–23. — DOI: 10.35330/1991-6639-2019-4-90-14-23. [in Russian]
11. Shkinderov M.S. Modelirovanie pomekhoustoichivosti sistemi kontrolya i upravleniya dostupom pri vozdeistvii elektrosticheskogo razryada [Modelling of noise immunity of access control and management systems under the influence of electrostatic discharge] / M.S. Shkinderov, R.R. Mubarakov // Trudi MAI [Proceedings of MAI]. — 2021. — № 120. — DOI: 10.34759/trd-2021-120-12. [in Russian]
12. Titov A.N. Vizualizatsiya danih v Python. Osnovi raboti s interaktivnoi bibliotekoi Altair : Uchebno-metodicheskoe posobie [Data visualisation in Python. Fundamentals of working with the Altair interactive library: Teaching and learning guide] / A.N. Titov, R.F. Tazieva. — Kazan: Kazan National Research Technological University, 2024. — 128 p. [in Russian]



13. Shurdilov I.S. Razrabotka sistemi raspoznavaniya emotsii po litsevim virazheniyam na osnove mashinnogo obucheniya [Development of a system for recognising emotions based on facial expressions using machine learning] / I.S. Shurdilov, M.G. Nuriev, M.G. Lapteva [et al.] // Mezhdunarodnii nauchno-issledovatel'skii zhurnal [International Research Journal]. — 2025. — № 6 (156). — DOI: 10.60797/IRJ.2025.156.52. [in Russian]
14. Lisenkova A.M. Sovremennye tekhnologii vizualizatsii upravlencheskikh reshenii [Modern technologies for visualising management decisions] / A.M. Lisenkova // Aktualnye aspekty modernizatsii rossiiskoi ekonomiki: X Vserossiiskaya zauchnaya nauchno-prakticheskaya konferentsiya studentov, aspirantov i molodikh uchenukh, Sankt-Peterburg, 25 dekabrya 2023 goda [Current aspects of modernising the Russian economy: 10th All-Russian Correspondence Scientific and Practical Conference for Students, Postgraduates and Young Scientists, Saint Petersburg, 25 December 2023]. — St.Petersburg: LETI, 2023. — P. 598–601. [in Russian]
15. Katasyov A.S. Neironechetkaya model i programmnyi kompleks avtomatizatsii formirovaniya nechetkikh pravil dlya otsenki sostoyaniya obektov [Neural fuzzy model and software package for automating the formation of fuzzy rules for assessing the condition of objects] / A.S. Katasyov // Avtomatizatsiya protsessov upravleniya [Automation of management processes]. — 2019. — № 1 (55). — P. 21–29. [in Russian]
16. Tirinov V.S. Obzor instrumentov dlya vizualizatsii daniikh [Overview of data visualisation tools] / V.S. Tirinov // Molodoi issledovatel: ot idei k proektu: Materialy VII studencheskoi nauchno-prakticheskoi konferentsii, Yoshkar-Ola, 24–28 aprelya 2023 goda [Young researcher: from idea to project: Proceedings of the VII Student Scientific and Practical Conference, Yoshkar-Ola, 24–28 April 2023] / Resp. ed. D.A. Mikheeva. — Yoshkar-Ola: Mari State University, 2023. — P. 138–143. [in Russian]
17. Shakirzyanov R.M. Metod avtomaticheskogo pozitsionirovaniya bespilotnikh apparatov na osnove raspoznavaniya signalnikh radialno-simmetrichnikh markerov podvodnikh tselei [Method for automatic positioning of unmanned aerial vehicles based on recognition of radial-symmetrical signal markers of underwater targets] / R.M. Shakirzyanov, M.P. Shleimovich, S.V. Novikova // Avtomatika i telemekhanika [Automation and Telemechanics]. — 2023. — № 7. — P. 93–120. — DOI: 10.31857/S0005231023070061. [in Russian]
18. Kornukov N.O. Obzor vozmozhnostei biblioteki D3 dlya vizualizatsii daniikh [Overview of the capabilities of the D3 library for data visualisation] / N.O. Kornukov // Molodezhnii nauchno-tekhnicheskii vestnik [Youth Scientific and Technical Bulletin]. — 2015. — № 8. — P. 12. [in Russian]
19. Smirnov Yu.N. Matematicheskaya model optimizatsii deyatelnosti dlya tsifrovoy sistemi upravleniya predpriyatiem [Mathematical model for optimising operations for a digital enterprise management system] / Yu.N. Smirnov, A.V. Kalyashina // Nauchno-tekhnicheskii vestnik Povolzhya [Scientific and Technical Bulletin of the Volga Region]. — 2023. — № 11. — P. 119–122. [in Russian]
20. Gibadullin R.F. Optimizatsiya asinkhronnikh operatsii v .NET [Optimisation of asynchronous operations in .NET] / R.F. Gibadullin, D.A. Gashigullin // Mezhdunarodnii nauchno-issledovatel'skii zhurnal [International Research Journal]. — 2025. — № 7 (157). — DOI: 10.60797/IRJ.2025.157.40. [in Russian]
21. Khabibullin F.F. Analiz parametrov oshibki polozheniya, peremeshcheniya idealnogo i realnogo mekhanizma dlya robototekhnicheskikh sistem [Analysis of position error parameters, movement of ideal and real mechanisms for robotic systems] / F.F. Khabibullin, R.T. Islamov, L.F. Khabibullina // Problemy mashinostroeniya i avtomatizatsii [Problems of Mechanical Engineering and Automation]. — 2024. — № 1. — P. 36–43. [in Russian]
22. Gnatenko Yu.A. Novye tekhnologii i instrumenty dlya vizualizatsii slozhnykh statisticheskikh daniikh [New technologies and tools for visualising complex statistical data] / Yu.A. Gnatenko, A.N. Dodarov // Tendentsii razvitiya nauki i obrazovaniya [Trends in the development of science and education]. — 2024. — № 110–17. — P. 76–83. — DOI: 10.18411/trnio-06-2024-932. [in Russian]
23. Guzhvenko Ye.I. Ispolzovanie kompleksa prakticheskikh zadaniy dlya vizualizatsii daniikh pri izuchenii voennosluzhashchimi distsipliny «Informatika i informatsionnye tekhnologii v professionalnoi deyatelnosti» [The use of a set of practical tasks for data visualisation in the study of the discipline ‘Computer Science and Information Technology in Professional Activity’ by military personnel] / Ye.I. Guzhvenko, V.Yu. Guzhvenko // Izvestiya Dagestanskogo gosudarstvennogo pedagogicheskogo universiteta. Psikhologo-pedagogicheskie nauki [Proceedings of the Dagestan State Pedagogical University. Psychological and Pedagogical Sciences]. — 2024. — Vol. 18. — № 2. — P. 37–41. — DOI: 10.31161/1995-0659-2024-18-2-37-41. [in Russian]