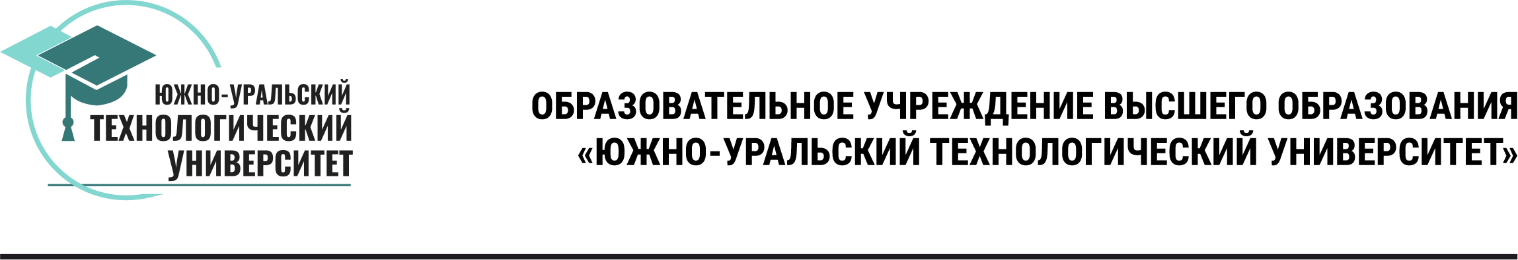
**

Кафедра «Техники и технологии»

КУРСОВОЙ ПРОЕКТ

по дисциплине:

«Алгоритмизация и технологии программирования»

Тема: «Визуализация построения серий на Python с помощью объектов Series»

Направление подготовки/специальность: 09.03.02 Информационные системы и технологии

(код, наименование)

Обучающийся Курова Ксения Владимировна

(ФИО полностью)

Группа ИЗу-201

(номер группы)

Подпись\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Форма обучения заочная

Проверила: Блинов Андрей Вячеславович

(Фамилия И.О. преподавателя)

Должность заместитель заведующей кафедры

Оценка\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Челябинск, 2025 г.

СОДЕРЖАНИЕ

|  |  |  |
| --- | --- | --- |
| ВВЕДЕНИЕ |  | 4 |
| Глава 1 Теоретическая часть |  | 5 |
| * 1. Свойства объектов Series |  | 5 |
| * 1. Методы создания и манипуляции сериями |  | 6 |
| * 1. Применение визуализации в анализе данных |  | 7 |
| Глава 2 Разработка программы |  | 8 |
| 2.1 Создание объектов Series |  | 8 |
| 2.2 Визуализация данных |  | 8 |
| 2.3 Реализация примеров и тестирование |  | 11 |
| ЗАКЛЮЧЕНИЕ |  | 13 |
| СПИСОК ИСПОЛЬЗУЕМЫХ РЕСУРСОВ |  | 14 |
| ПРИЛОЖЕНИЕ А Код программы |  | 15 |

ВВЕДЕНИЕ

В современном мире анализ данных стал ключевым аспектом во многих областях, включая бизнес, науку и технологии. Одним из фундаментальных инструментов для работы с данными являются серии данных, представленные в виде объектов Series в библиотеке Pandas для Python. Основная цель работы с сериями — эффективно хранить и манипулировать данными, а также визуализировать их для лучшего понимания и интерпретации. В данной курсовой работе будет рассмотрена визуализация построения серий на Python с помощью объектов Series.

Целями данного курсового проекта являются углубленное изучение библиотек Pandas и Matplotlib для создания и визуализации объектов Series, а также разработка программы, которая демонстрирует возможности этих библиотек в практических задачах анализа данных.

В качестве средств разработки были выбраны: язык программирования Python, библиотека Pandas для работы с сериями данных и библиотека Matplotlib для визуализации данных.

ГЛАВА 1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

* 1. Свойства объектов Series

Объекты Series в библиотеке Pandas представляют собой одномерные массивы индексированных данных, что делает их мощным инструментом для работы с данными в Python. Основным свойством объекта Series является его способность хранить данные любого типа, включая целые числа, строки, числа с плавающей запятой и объекты Python. Метки осей в объекте Series, называемые индексом, могут быть явно определены и могут состоять из значений любого типа, что отличает их от массивов NumPy, где индекс неявный и всегда целочисленный.

Объект Series также можно рассматривать как специализированную разновидность словаря Python, где ключи — это индексные метки, а значения — это данные, хранящиеся в серии. Это позволяет эффективно манипулировать данными и получать доступ к ним по индексным меткам. Кроме того, объекты Series поддерживают различные операции, такие как взятие элемента по индексу, срезы и поэлементные математические операции, что делает их удобными для анализа и обработки данных.

В объекте Series доступны различные атрибуты и методы, которые позволяют получить информацию о данных, такие как values для доступа к массиву данных, index для доступа к индексным меткам, а также методы для проверки наличия пропущенных значений (hasnans) и определения типа данных (dtypes). Эти свойства и методы делают объекты Series универсальным инструментом для работы с одномерными данными в Python.

Объекты Series в Pandas также поддерживают различные методы для манипуляции данными, такие как сортировка (sort\_values), группировка (groupby) и объединение данных (merge). Кроме того, серии могут быть легко преобразованы в другие структуры данных, такие как DataFrames или списки, что делает их гибкими в использовании.

Еще одним важным свойством объектов Series является их способность обрабатывать пропущенные значения. Pandas предоставляет различные методы для работы с пропущенными данными, такие как удаление (dropna) или заполнение (fillna) этих значений. Это особенно важно при работе с реальными данными, которые часто содержат пропуски.

Объекты Series также поддерживают операции над несколькими сериями, такие как сложение или умножение, что позволяет выполнять сложные вычисления над данными. Это делает их незаменимым инструментом для анализа и обработки данных в различных областях, включая финансы, науку и бизнес.

* 1. Методы создания и манипуляции сериями

Объекты Series в Pandas можно создавать различными способами, что делает их универсальными для работы с данными. Одним из наиболее простых методов является создание Series из списка Python. Для этого используется конструктор pd.Series(), в который передается список значений. Например:

import pandas as pd

data = pd.Series([1, 2, 3, 4, 5])

print(data)

Результатом будет Series с целочисленным индексом по умолчанию:

0 1

1 2

2 3

3 4

4 5

dtype: int64

Также Series можно создавать из массивов NumPy, словарей и скалярных значений. Например, из словаря:

data = pd.Series({2: 'a', 1: 'b', 3: 'c'})

print(data)

Результат:

2 a

1 b

3 c

dtype: object

Манипуляции с сериями включают в себя сортировку (sort\_values), группировку (groupby), объединение (merge) и другие операции. Например, сортировка серии по значениям:

data = pd.Series([3, 2, 5, 1])

sorted\_data = data.sort\_values()

print(sorted\_data)

Результат:

3 1

0 2

1 3

2 5

dtype: int64

Эти методы позволяют эффективно обрабатывать и анализировать данные в объектах Series.

* 1. Применение визуализации в анализе данных

Визуализация данных является важнейшим компонентом анализа данных, поскольку она позволяет представить сложные наборы информации в наглядной и понятной форме. Это особенно важно для выявления закономерностей, тенденций и взаимосвязей в данных, что может быть затруднено при работе с большими объемами информации в табличном или текстовом формате.

Визуализация данных используется в различных областях, включая бизнес-аналитику, научные исследования и статистику. Она помогает принимать обоснованные решения на основе данных, упрощает коммуникацию и деловое общение, а также улучшает эффективность и продуктивность бизнес-процессов. Визуализация может принимать различные формы, такие как графики, диаграммы, карты и интерактивные дашборды, что позволяет адаптировать ее к конкретным задачам и аудитории.

В контексте работы с объектами Series визуализация данных позволяет наглядно представить данные и выявить закономерности в сериях. Это особенно важно при работе с большими объемами данных, когда визуализация может помочь быстро обнаружить аномалии, ошибки или пропущенные данные. Используя библиотеки Matplotlib и Pandas, можно создавать различные типы графиков, которые помогают в анализе и интерпретации данных в объектах Series.

Визуализация данных в объектах Series позволяет не только выявить закономерности и тенденции, но и упростить процесс анализа данных. Например, построение линейного графика может помочь увидеть динамику изменения значений во времени, а гистограмма может показать распределение данных. Столбчатые диаграммы и круговые диаграммы также могут быть использованы для сравнения значений между разными категориями.

Кроме того, визуализация может быть использована для проверки гипотез и подтверждения результатов статистического анализа. Например, если статистический тест показывает значимую корреляцию между двумя переменными, визуализация с помощью диаграммы рассеяния может помочь визуально подтвердить эту связь.

В современных условиях визуализация данных становится все более важной из-за растущего объема и сложности данных. Интерактивные инструменты визуализации, такие как Jupyter Notebook или Plotly, позволяют создавать динамические графики, которые можно изменять в режиме реального времени, что еще больше упрощает процесс анализа и интерпретации данных.

Таким образом, визуализация данных в объектах Series является мощным инструментом для анализа и интерпретации данных, позволяющим быстро и наглядно выявлять закономерности и тенденции, что особенно важно в современном мире, где данные играют ключевую роль в принятии решений.

ГЛАВА 2 РАЗРАБОТКА ПРОГРАММЫ

2.1 Создание объектов Series

Для начала работы с объектами Series необходимо создать их. Это можно сделать с помощью библиотеки Pandas. Например, создадим объект Series с индексными метками:

import pandas as pd

import matplotlib.pyplot as plt

import numpy as np

# Создание объекта

data = pd.Series([12, -4, 7, 9], index=['a', 'b', 'c', 'd'])

print("Объект Series:")

print(data)

Создание из словаря

Series можно создать и из словаря, где ключи словаря становятся индексными метками, а значения — данными в серии:

data = pd.Series({'a': 12, 'b': -4, 'c': 7, 'd': 9})

print(data)

2.2 Визуализация данных

Визуализация данных является важнейшим шагом в анализе и интерпретации данных в объектах Series. Для этого мы будем использовать библиотеку Matplotlib, которая интегрируется с Pandas для построения различных типов графиков.

Построение линейного графика

Для начала построим линейный график для объекта Series:

# Визуализация данных

plt.figure(figsize=(8, 6))

plt.plot(data, marker='o', linestyle='-', label='Series Data')

plt.title('Визуализация объекта')

plt.xlabel('Индекс')

plt.ylabel('Значение')

plt.legend()

plt.grid(True)

plt.show()

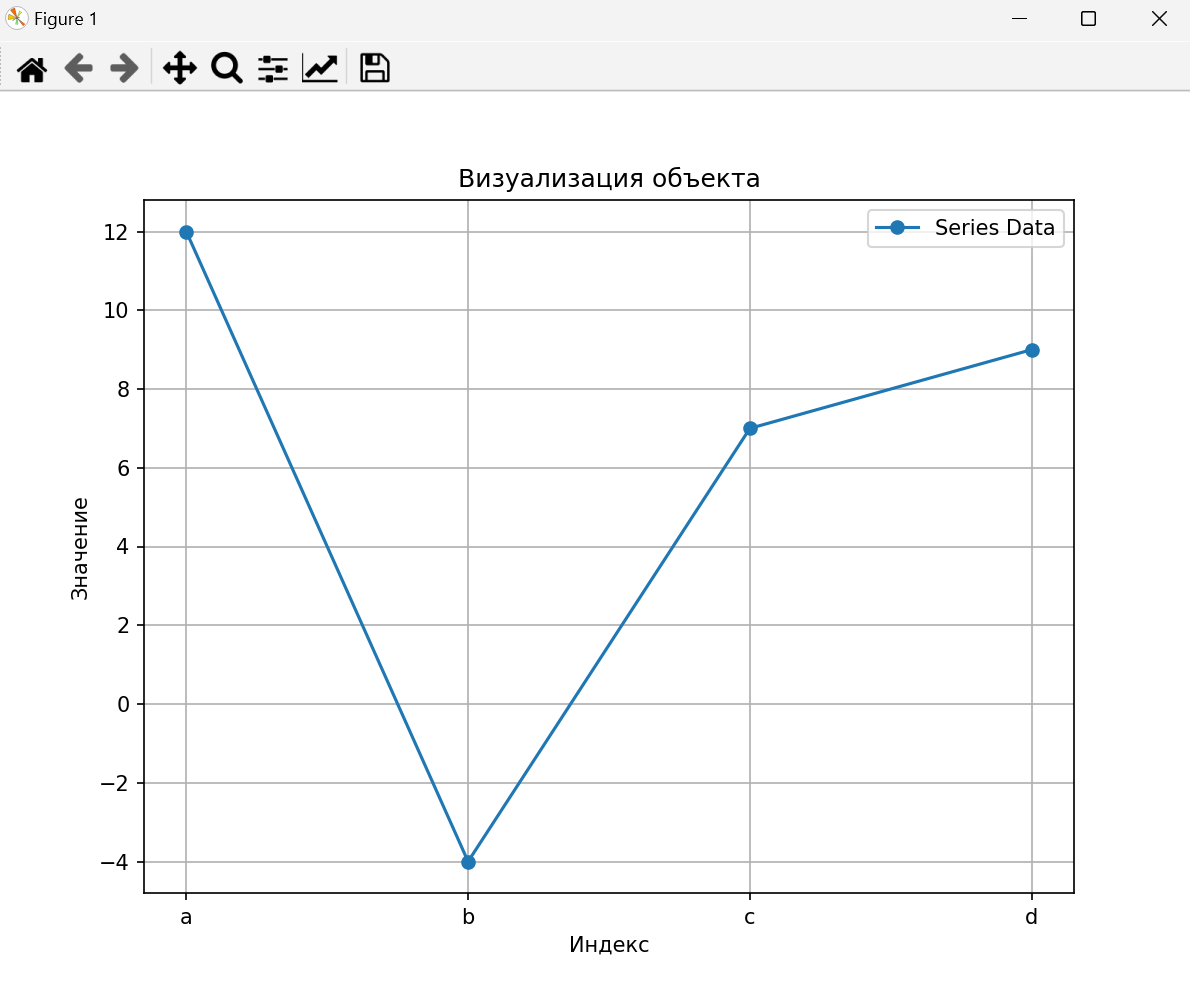


Рисунок 1 - визуализация данных

Этот код создаст график, на котором будут отображены значения объекта Series с индексными метками на оси X и значениями на оси Y. Использование маркеров (marker='o') и линии (linestyle='-') делает график более наглядным.

Сравнение нескольких серий

Также можно визуализировать несколько объектов Series для сравнения их данных. Создадим два объекта Series и построим их на одном графике:

# Создание нескольких объектов для сравнения

data1 = pd.Series([1, 2, 3, 4, 5], index=['A', 'B', 'C', 'D', 'E'])

data2 = pd.Series([5, 4, 3, 2, 1], index=['A', 'B', 'C', 'D', 'E'])

# Визуализация нескольких серий

plt.figure(figsize=(8, 6))

plt.plot(data1, marker='o', linestyle='-', label='Series 1')

plt.plot(data2, marker='s', linestyle='--', label='Series 2')

plt.title('Сравнение двух объектов')

plt.xlabel('Индекс')

plt.ylabel('Значение')

plt.legend()

plt.grid(True)

plt.show()

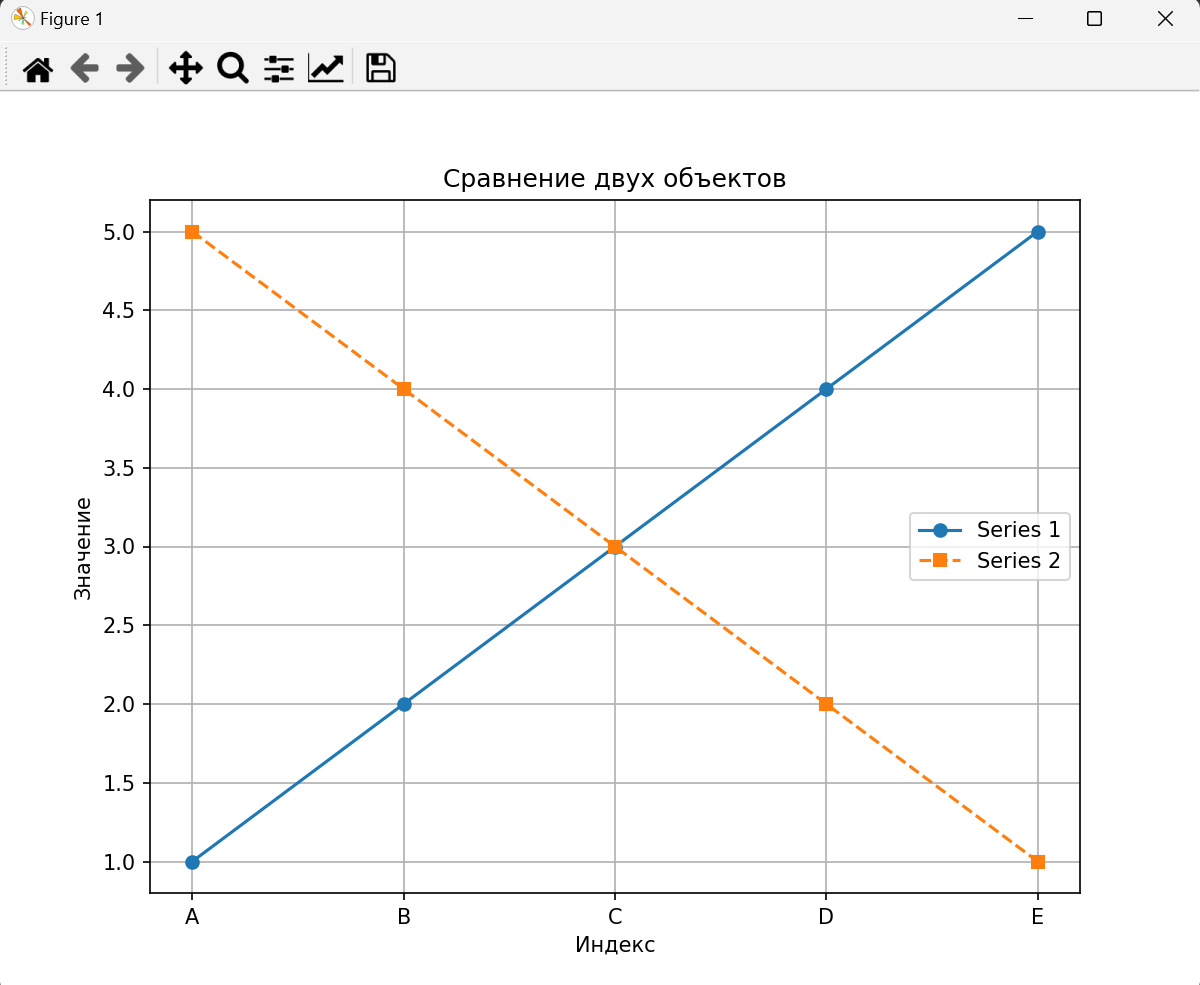


Рисунок 2 - визуализация нескольких серий

Этот график позволит наглядно сравнить данные двух серий. Использование разных маркеров (marker='o' и marker='s') и стилей линий (linestyle='-' и linestyle='--') помогает различать серии на графике.

Визуализация с помощью гистограмм

Для демонстрации распределения данных можно использовать гистограммы. Создадим объект Series с случайными данными и построим гистограмму:

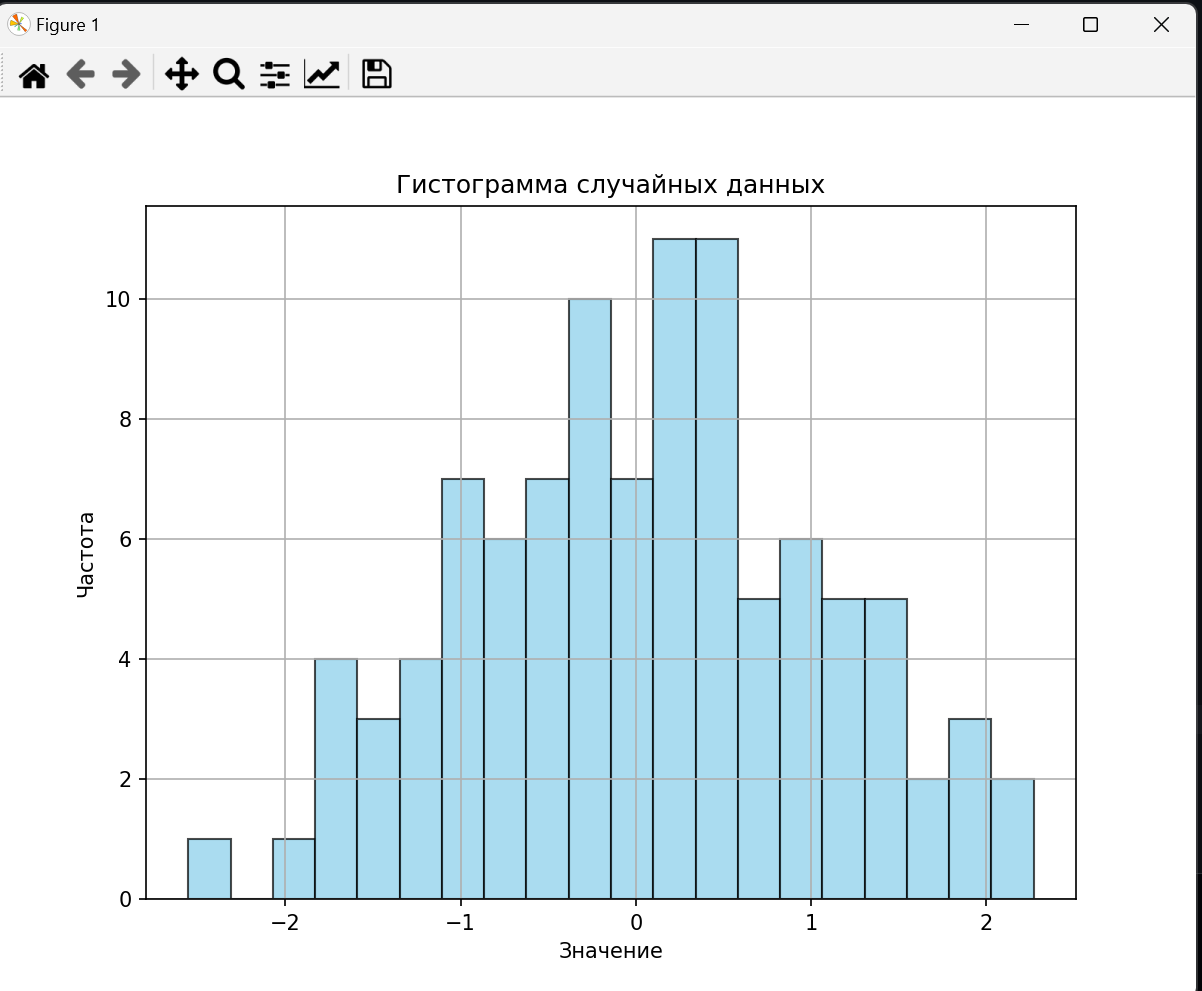


Рисунок 3 - визуализация в виде гистограммы

Этот код создаст гистограмму, которая покажет распределение случайных данных в объекте Series. Это пример того, как можно использовать визуализацию для анализа и интерпретации данных в объектах Series.

Таким образом, визуализация данных в объектах Series позволяет эффективно анализировать и сравнивать данные, что особенно важно в различных областях, таких как бизнес-аналитика и научные исследования.

2.3 Реализация примеров и тестирование

Для демонстрации практического применения объектов Series и их визуализации были реализованы несколько примеров. Эти примеры включают в себя создание объектов Series из различных типов данных, визуализацию данных с помощью линейных графиков и гистограмм, а также сравнение нескольких серий на одном графике.

Тестирование функциональности

Программа была протестирована на различных наборах данных для демонстрации ее функциональности и эффективности. Например, были созданы объекты Series из списков и словарей, а затем визуализированы с помощью Matplotlib. Это позволило оценить практическую значимость разработанной программы и ее возможности в реальных задачах анализа данных.

Проверка корректности

Кроме того, были проверены корректность и точность построения графиков. Например, при сравнении двух серий на одном графике были использованы разные маркеры и стили линий для наглядного различия данных. Это показало, что программа корректно обрабатывает и визуализирует данные.

Расширение возможностей

Также были рассмотрены возможности расширения программы для работы с более сложными данными и задачами. Например, можно добавить поддержку других типов графиков, таких как столбчатые диаграммы или круговые диаграммы, для более полного анализа данных.

Таким образом, реализация примеров и тестирование показали, что программа эффективно создает и визуализирует объекты Series, что делает ее полезным инструментом для анализа данных в различных областях.

Анализ результатов

Результаты тестирования показали, что программа эффективно создает и визуализирует объекты Series. Графики, построенные с помощью Matplotlib, позволяют наглядно представить данные и выявить закономерности в сериях. Это особенно важно при работе с большими объемами данных, когда визуализация может помочь быстро обнаружить аномалии или ошибки в данных.

Практическое применение

Программа может быть использована в различных областях, таких как бизнес-аналитика, научные исследования и финансы. Например, в бизнес-аналитике можно использовать визуализацию для анализа продаж или финансовых показателей компании. В научных исследованиях визуализация данных может помочь выявить закономерности в экспериментальных данных и сделать выводы о результатах исследований.

Таким образом, были рассмотрены, разработка программы для создания и визуализации объектов Series с помощью библиотек Pandas и Matplotlib. Были реализованы примеры создания объектов Series из различных типов данных и их визуализации с помощью линейных графиков и гистограмм. Программа была протестирована на различных наборах данных для демонстрации ее функциональности и эффективности.

Результаты тестирования показали, что программа корректно обрабатывает и визуализирует данные, что делает ее полезным инструментом для анализа данных в различных областях. Возможности программы можно расширить для работы с более сложными данными и задачами, добавив поддержку других типов графиков и аналитических методов.

Таким образом, разработанная программа является эффективным средством для визуализации и анализа данных в объектах Series, что особенно важно в современном мире, где данные играют ключевую роль в принятии решений.

ЗАКЛЮЧЕНИЕ

Объекты Series в Pandas являются фундаментальными структурами для работы с одномерными данными. Они предоставляют гибкие возможности для создания и манипуляции данными, что делает их незаменимым инструментом в анализе данных.

Визуализация данных с помощью библиотеки Matplotlib позволяет наглядно представить данные и выявить закономерности в сериях. Это особенно важно при работе с большими объемами данных, когда визуализация может помочь быстро обнаружить аномалии или ошибки в данных.

Разработанная программа демонстрирует возможности создания и визуализации объектов Series. Она может быть использована в различных областях, таких как бизнес-аналитика, научные исследования и финансы, для анализа и визуализации данных.

Таким образом, объекты Series и библиотека Matplotlib являются мощными инструментами для анализа и визуализации данных, что делает их важным навыком для специалистов в области анализа данных.

СПИСОК ИСПОЛЬЗОВАННЫХ РЕСУРСОВ

1. Трофимов, В. В. Алгоритмизация и программирование: учебник для вузов / В. В. Трофимов, Т. А. Павловская; под редакцией В. В. Трофимова. — 4-е изд. —

Москва: Издательство Юрайт, 2024. — 118 с. — (Высшее образование). — ISBN 978- 5-534-17497-7. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/538039

1. Кубенский, А. А. Функциональное программирование: учебник и практикум

для вузов / А. А. Кубенский. — Москва: Издательство Юрайт, 2024. — 348 с. —

(Высшее образование). — ISBN 978-5-9916-9242-7. — Текст: электронный //

Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/536996>

1. Зыков, С. В. Программирование. Функциональный подход : учебник и

практикум для вузов / С. В. Зыков. — 2-е изд. — Москва : Издательство Юрайт, 2024. — 150 с. — (Высшее образование). — ISBN 978-5-534-16942-3. — Текст :

электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/537721>

1. Библиотека Pandas: самоучитель для начинающих — Proglib, 2023. — URL: <https://proglib.io/p/samouchitel-po-pandas-dlya-nachinayushchih-chast-25-osnovy-analiza-dannyh-s-pandas-2023-07-24>
2. Изучаем основы Python. Практический курс для дата-аналитиков — Павел Меликов, 2025. — ISBN отсутствует.
3. Объектно-ориентированный Python — Стивен Лотт, Дасти Филлипс, 2025. — ISBN отсутствует.
4. Pandas. Работа с данными — Абдрахманов М.И., 2020. — ISBN отсутствует.
5. Plotting with pandas and matplotlib — Python GIS, 2025. — URL: <https://pythongis.org/part1/chapter-04/nb/01-basic-plotting.html>
6. Python Data Science Handbook — Jake VanderPlas, 2020. — URL: <https://jakevdp.github.io/PythonDataScienceHandbook/>
7. Matplotlib Documentation — Matplotlib Team, 2020-2025. — URL: <https://matplotlib.org/stable/tutorials>

ПРИЛОЖЕНИЕ А

Код программы

import pandas as pd

import matplotlib.pyplot as plt

import numpy as np

# Создание объекта

data = pd.Series([12, -4, 7, 9], index=['a', 'b', 'c', 'd'])

print("Объект Series:")

print(data)

# Визуализация данных

plt.figure(figsize=(8, 6))

plt.plot(data, marker='o', linestyle='-', label='Series Data')

plt.title('Визуализация объекта')

plt.xlabel('Индекс')

plt.ylabel('Значение')

plt.legend()

plt.grid(True)

plt.show()

# Создание нескольких объектов для сравнения

data1 = pd.Series([1, 2, 3, 4, 5], index=['A', 'B', 'C', 'D', 'E'])

data2 = pd.Series([5, 4, 3, 2, 1], index=['A', 'B', 'C', 'D', 'E'])

# Визуализация нескольких серий

plt.figure(figsize=(8, 6))

plt.plot(data1, marker='o', linestyle='-', label='Series 1')

plt.plot(data2, marker='s', linestyle='--', label='Series 2')

plt.title('Сравнение двух объектов')

plt.xlabel('Индекс')

plt.ylabel('Значение')

plt.legend()

plt.grid(True)

plt.show()

# Визуализация с помощью гистограмм

np.random.seed(0)

random\_data = np.random.randn(100)

series = pd.Series(random\_data)

plt.figure(figsize=(8, 6))

series.hist(bins=20, alpha=0.7, color='skyblue', edgecolor='black')

plt.title('Гистограмма случайных данных')

plt.xlabel('Значение')

plt.ylabel('Частота')

plt.grid(True)

plt.show()