

УДК 004.423

DOI 10.51691/2541-8327_2023_6_10

ВИЗУАЛИЗАЦИЯ ДАННЫХ С ПОМОЩЬЮ БИБЛИОТЕК PANDAS И MATPLOTLIB ДЛЯ ЯЗЫКА ПРОГРАММИРОВАНИЯ PYTHON

Дрянкова Д.А.

*студент факультета информатики и вычислительной техники,
Хакасский государственный университет имени Н.Ф. Катанова,
г. Абакан, Россия¹*

Аннотация: Целью данного исследования являлась визуализация данных из таблицы Pandas с использованием библиотеки Matplotlib.

В ходе исследования были рассмотрены различные типы графиков, такие как линейные графики, гистограммы, круговые диаграммы и диаграммы разброса. Были приведены примеры кода, демонстрирующие использование этих методов для визуализации данных.

Результаты исследования показали, что библиотека Pandas с интеграцией Matplotlib предоставляет мощные инструменты для визуализации данных. Мы смогли наглядно представить тренды, распределения и отношения между данными.

Визуализация данных позволяет выявить закономерности, обнаружить выбросы и сделать выводы на основе визуального анализа данных.

Ключевые слова: Python, Pandas, Matplotlib, фреймы данных, таблицы данных, таблицы, визуализация, график, диаграмма.

DATA VISUALIZATION USING PANDAS AND MATPLOTLIB LIBRARIES FOR THE PYTHON PROGRAMMING LANGUAGE

¹ Научный руководитель: Замулин И.С. заведующий кафедрой ПОВТиАС, Хакасский государственный университет имени Н.Ф. Катанова, г. Абакан, Россия

Dryakova D.A.

*student of the Faculty of Computer Science and Computer Engineering,
N.F. Katanov Khakass State University,
Abakan, Russia*

Abstract: The purpose of this study was to visualize data from the Pandas table using the Matplotlib library.

In the course of the study, various types of graphs were considered, such as line graphs, histograms, pie charts and scatter charts. Code examples were provided to demonstrate the use of these methods for data visualization.

The results of the study showed that the Pandas library with Matplotlib integration provides powerful tools for data visualization. We were able to visualize trends, distributions, and relationships between data.

Data visualization allows you to identify patterns, detect outliers and draw conclusions based on visual data analysis.

Keywords: Python, Pandas, Matplotlib, DataFrames, data tables, tables, visualization, graph, diagram.

Визуализация данных в библиотеке Pandas основана на интеграции с другой мощной библиотекой визуализации данных в Python, такой как Matplotlib. Pandas предоставляет удобные методы для создания графиков и диаграмм на основе данных, хранящихся в таблицах DataFrame [1].

Когда вызывается метод визуализации, Pandas создает объекты графиков с использованием Matplotlib и передает ему данные для отображения. Методы визуализации Pandas предоставляют удобные интерфейсы для создания различных типов графиков, указания параметров стиля, цветов, меток и других атрибутов графика [2].

Процесс визуализации в Pandas обычно включает следующие шаги:

1) Создание объекта DataFrame: Исходные данные хранятся в объекте DataFrame, который представляет собой таблицу с метками строк и столбцов.

2) Выбор данных: Из DataFrame выбираются необходимые столбцы или строки для визуализации. Это может быть выполнено с помощью индексации или фильтрации данных.

3) Вызов метода визуализации: Вызывается соответствующий метод визуализации Pandas, такой как *plot()*, *hist()*, *scatter()*, *bar()*, и т.д., с указанием необходимых параметров.

4) Отображение графика: Pandas передает данные визуализации Matplotlib, который отображает график согласно заданным параметрам. Matplotlib обрабатывает создание фигуры, осей, меток, легенды и других элементов графика.

5) Настройка графика: Pandas предоставляет методы для настройки внешнего вида графика, такие как установка заголовков, меток осей, изменение цветов, стилей линий и других атрибутов.

6) Отображение графика: Итоговый график отображается в окне или сохраняется в файл, в зависимости от настроек.

Визуализация данных является важной частью анализа данных, и библиотека Pandas предоставляет различные методы для визуализации данных, хранящихся в таблицах. Ниже приведены некоторые из наиболее распространенных методов визуализации данных в Pandas:

1) Гистограммы (histograms): Гистограммы представляют собой визуализацию распределения данных. Метод *hist()* позволяет построить гистограмму для одного или нескольких столбцов в таблице (Рис. 1).

```
[9]: import pandas as pd
import matplotlib.pyplot as plt

# Создание таблицы с данными
data = {'Возраст': [25, 26, 21, 43, 30, 44, 22, 26, 40,
                    30, 28, 35, 22, 29, 32, 40, 45, 37]}
df = pd.DataFrame(data)

# Построение гистограммы
plt.hist(df['Возраст'], bins=10)
plt.xlabel('Возраст')
plt.ylabel('Счёт')
plt.title('Распределение по возрасту')
plt.show()
```

Рисунок 1 - Создание таблицы данных и построение гистограммы
[разработано автором]

Итоговый график получившейся гистограммы (Рис. 2).

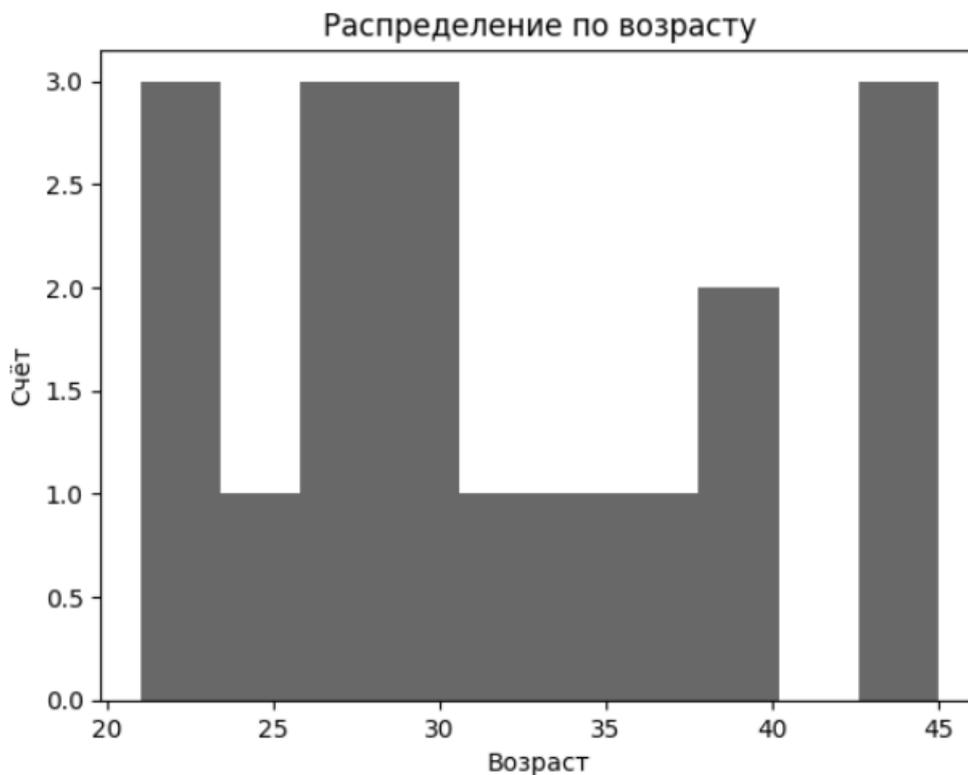


Рисунок 2 – Результат построения гистограммы [разработано автором]

2) Круговые диаграммы (pie charts): Круговые диаграммы позволяют визуализировать доли различных категорий данных. Метод *pie()* используется для создания круговой диаграммы (Рис. 3).

```
[3]: import pandas as pd
import matplotlib.pyplot as plt

# Создание таблицы с данными
data = {'Категория': ['A', 'B', 'C', 'D'],
        'Значение': [25, 30, 15, 30]}
df = pd.DataFrame(data)

# Построение круговой диаграммы
plt.pie(df['Значение'], labels=df['Категория'], autopct='%1.1f%%')
plt.title('Распределение по категориям')
plt.show()
```

Рисунок 3 - Создание таблицы данных и построение круговой диаграммы
[разработано автором]

Итоговый график получившейся круговой диаграммы (Рис. 4).

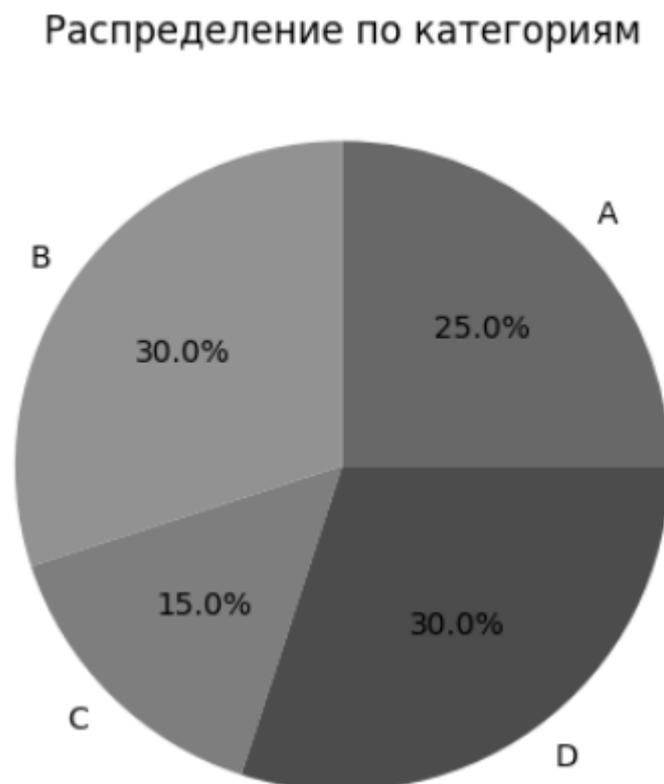


Рисунок 4 – Результат построения круговой диаграммы [разработано автором]

3) Линейные графики (line plots): Линейные графики позволяют визуализировать изменение данных во времени. Метод *plot()* используется для создания линейного графика (Рис. 5).

```
[1]: import pandas as pd
import matplotlib.pyplot as plt

# Создание таблицы с данными
data = {'Год': [2010, 2011, 2012, 2013, 2014],
        'Продажи': [100, 120, 90, 110, 150]}
df = pd.DataFrame(data)

# Построение линейного графика
plt.plot(df['Год'], df['Продажи'])
plt.xlabel('Год')
plt.ylabel('Продажи')
plt.title('Тенденция продаж')
plt.show()
```

Рисунок 5 – Создание таблицы данных и построение линейного графика
[разработано автором]

Итоговый линейный график (Рис. 6).



Рисунок 6 – Результат построения линейного графика [разработано автором]

4) Диаграммы разброса (scatter plots): Диаграммы разброса позволяют визуализировать связь между двумя переменными. Метод `scatter()` используется для создания диаграммы разброса (Рис. 7).

```
[2]: import pandas as pd
import matplotlib.pyplot as plt

# Создание таблицы с данными
data = {'Вес': [50, 60, 70, 80, 90],
        'Рост': [160, 165, 170, 175, 180]}
df = pd.DataFrame(data)

# Построение диаграммы разброса
plt.scatter(df['Вес'], df['Рост'])
plt.xlabel('Вес')
plt.ylabel('Рост')
plt.title('Соотношение веса и роста')
plt.show()
```

Рисунок 7 - Создание таблицы данных и построение диаграммы разброса

[разработано автором]

Итоговый график получившейся диаграммы (Рис. 8).

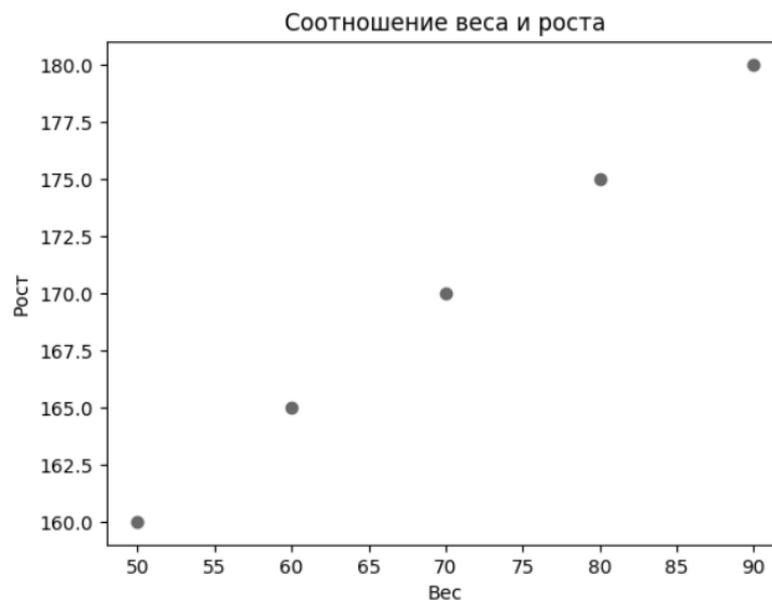


Рисунок 8 – Результат построения диаграммы разброса [разработано автором]

Это лишь некоторые из примеров визуализации данных с помощью Pandas и Matplotlib. Обратите внимание, что Matplotlib предоставляет множество функций для настройки внешнего вида графиков, таких как добавление легенды, изменение цветов, стилей линий и других атрибутов [3].

Заключение

В заключении можно прийти к выводу, что Pandas упрощает процесс визуализации данных, предоставляя высокоуровневый интерфейс и интеграцию с Matplotlib. Это позволяет исследователям данных и аналитикам быстро создавать графики и диаграммы для визуализации и анализа данных.

Библиографический список

1. Pandas [Электронный ресурс] [pydata.org](https://www.pydata.org/pandas/) — Режим доступа — URL: https://www.pydata.org/pandas/docs/user_guide/visualization.html.
2. Python 3 [Электронный ресурс] python-scripts.com — Режим доступа — URL: <https://python-scripts.com/plot-with-pandas>
3. Артем Груздев. Предварительная подготовка данных в Python. Том 1. Инструменты и валидация [Текст] / А. В. Груздев. – М.: ДМК-Пресс, 2023 г. – 816 с.

Оригинальность 78%