6. Лабораторна робота. Застосування бібліотеки Pandas до задач аналізу даних

Мета: засвоїти можливості роботи бібліотеки Pandas до аналізу та обробки даних.

Теоретичні відомості та методичні рекомендації

Бібліотека Pandas – це одна з найпопулярніших бібліотек для

Pandas – це високорівнева бібліотека Python для аналізу та обробки даних. Вона забезпечує зручні структури даних і функції для роботи з табличними даними, подібними до електронних таблиць (Excel) або реляційних баз даних.

Основні можливості Pandas:

1. Основні структури даних:
   - Series – одномірний масив, схожий на стовпчик у таблиці або список;
   - DataFrame – двовимірна таблиця з рядками та стовпцями (основний об'єкт Pandas, який є табличною структурою даних).
2. Імпорт та експорт даних:
   - зчитування даних із csv, Excel, sql, json;
   - запис у файли різних форматів.
3. Обробка та маніпуляції з даними:
   - фільтрація, сортування, групування, агрегування;
   - обробка відсутніх значень.
4. Аналіз даних та статистика:
   - обчислення середнього, медіани, стандартного відхилення тощо;
   - робота з часовими рядами.
5. Візуалізація:
   - інтеграція з бібліотекою Matplotlib для побудови графіків.

Наприклад, створимо DataFrame, у якому міститься така інформація: ім'я, вік і оцінка студента. Треба вивести студентів, вік яких більше 22 та обчислити середню оцінку студентів.

```
import pandas as pd

# Створення DataFrame
data = {
    "Ім'я": ["Анна", "Богдан", "Віктор"],
    "Вік": [23, 25, 22],
    "Оцінка": [90, 85, 88]
}

df = pd.DataFrame(data)

# Виведення таблиці
print(df)
```

```
# Вибір рядків за умовою
print(df[df["Вік"] > 22])

# Обчислення середньої оцінки
print("Середня оцінка:", df["Оцінка"].mean())
```

У результаті отримаємо:

```
Ім'я  Вік  Оцінка
0    Анна   23      90
1  Богдан   25      85
2  Віктор   22      88
     Ім'я  Вік  Оцінка
0    Анна   23      90
1  Богдан   25      85
Середня оцінка: 87.66666666666667
```

Бібліотека Pandas розроблялася для роботи з табличними даними. Популярними типами файлів для їх зберігання є:

- csv – текстовий формат, у якому значення в стовпцях відокремлені один від одного роздільником, часто комою;
- xlsx або xls – формати файлів електронних таблиць Microsoft Excel;
- json-файли – це текстовий формат, призначений для зберігання структурованих даних.

Для читання та запису таблиць зазначених форматів у Pandas існують спеціальні методи:

- для читання файлів csv використовується метод `pd.read_csv()`;
- для читання файлів xlsx використовується метод `pd.read_excel()`;
- для читання файлів json використовується метод `pd.read_json()`;
- для читання бази даних SQL використовується метод `pd.read_sql()`;
- для читання HTML-таблиць використовується метод `pd.read_html()`;
- для читання великих файлів використовується метод `pd.read_parquet()`;
- для читання ZIP-архівів використовується метод з додатковим параметром `pd.read_csv("data.zip", compression="zip")`.

Метод `pd.read_csv("data.csv")` – найпоширеніший метод для завантаження даних у Pandas. Корисні параметри:

- `sep=";"` – змінює роздільник (якщо не кома);
- `header=None` – якщо немає заголовків;
- `usecols=["Ім'я", "Вік"]` – читає лише вибрані стовпці;
- `encoding="utf-8"` – задає кодування (важливо для українських текстів);

– `dtype={"Оцінка": float}` – задає тип даних для стовпця;
– `parse_dates=["Дата"]` – автоматично конвертує стовпець у формат дати.

У Pandas є багато методів для перегляду та аналізу DataFrame.

1. Основні методи для перегляду DataFrame.

Виведення перших або останніх рядків:

– `df.head(n)` – показує перші n рядків (за замовчуванням 5);
– `df.tail(n)` – показує останні n рядків.

Основна інформація про DataFrame:

– `df.info()` – загальна інформація про стовпці, типи даних та пропущені значення;
– `df.shape` – розмірність DataFrame (кількість рядків і стовпців);
– `df.columns` – список назв стовпців;
– `df.index` – інформація про індекси.

Перегляд статистичних характеристик:

– `df.describe()` – основна статистика (середнє, медіана, мін., макс.);
– `df.describe(include="all")` – статистика для всіх типів даних (не лише числових);
– `df["Оцінка"].mean()` – середнє значення конкретного стовпця.

Вибір випадкових рядків: `df.sample(n)` – випадковий вибір n рядків.

2. Перегляд вмісту DataFrame.

Вибір конкретного стовпця:

– `df["Ім'я"]` – вибір одного стовпця;
– `df[["Ім'я", "Оцінка"]]` – вибір кількох стовпців.

Вибір рядків за номером (iloc):

– `df.iloc[0]` – перший рядок;
– `df.iloc[1:4]` – рядки з 1 по 3 (не включаючи 4).

Вибір рядків за умовою:

– `df[df["Вік"] > 22]` – вибір рядків, де "Вік" більше 22;
– `df[df["Ім'я"] == "Анна"]` – вибір рядків, де ім'я "Анна".

Перевірка наявності пропущених значень: `df.isnull().sum()` – кількість пропущених значень у кожному стовпці.

3. Методи для роботи зі структурою DataFrame.

Перейменування стовпців, наприклад:
`df.rename(columns={"Оцінка": "Середній бал"}, inplace=True)`

Сортування, наприклад:

– `df.sort_values("Вік")` – сортує за віком (за зростанням).
– `df.sort_values("Вік", ascending=False)` – сортує за спаданням.

Видалення стовпців або рядків, наприклад:

– `df.drop(columns=["Оцінка"])` – видаляє стовпець.

– `df.drop(index=[0, 2])` – видаляє рядки за індексом.

Таким чином, бібліотека Pandas спрощує обробку даних, має інтуїтивно зрозумілий синтаксис, добре інтегрується з іншими бібліотеками Python (NumPy, Matplotlib, Seaborn), підходить для великих обсягів даних.

Завдання до лабораторної роботи

1. Завантажити з https://www.kaggle.com/datasets Public Datasets (csv-файл з даними). Провести аналіз даних:

1) Перетворити csv-дані в об'єкт DataFrame.

2) Вивести на екран початок та кінець файлу.

3) Вивести на екран останні 15 рядків файлу.

4) Вивести на екран декілька довільних стовпчиків.

5) Додати новий стовпчик Total і присвоїти йому суму або кількість деяких значень з інших стовпчиків.

6) Перейменувати колонки, які містять два і більше слів.

7) Зробити перезапис DataFrame.

2. За допомогою методу `read_html()` виконайте парсінг таблички з вебсторінки, наприклад, з офіційної сторінки українського банку з курсом валют.

1) Визначить кількість отриманих таблиць.

2) Отримайте DataFrame з курсами валют.

3. Використовуючи базові фільтри, відфільтруйте дані, за числовими даними. Збережіть отриманий результат у інший csv-файл з даними. Зробіть висновки.

4. Виконайте розділення csv-файлу на частини (по 15 рядків) та об'єднайте отримані частини в один DataFrame.

Питання для самоконтролю

1. Що представляє собою у бібліотеці Pandas об'єкт Series?
2. Що представляє собою у бібліотеці Pandas об'єкт DataFrame?
3. Які існують методи DataFrame читання тестових форматів?
4. Розкажіть про основні методи для перегляду DataFrame.
5. Розкажіть про способи перегляду вмісту DataFrame.
6. Охарактеризуйте методи для роботи зі структурою DataFrame.