TP: Data - File Handeling - Pandas

1 Exercise 1: Data File Operations

Objective

Practice handling different file formats (CSV and JSON) using Python's built-in libraries and Pandas.

1. Create a student records management system with the following data:

```
students = [
    {'id': '001', 'name': 'John Doe',
     'grade': 85, 'courses': ['Math', 'Physics']},
    {'id': '002', 'name': 'Jane Smith',
     'grade': 92, 'courses': ['Chemistry', 'Biology']},
    {'id': '003', 'name': 'Bob Wilson',
     'grade': 78, 'courses': ['Physics', 'Math']}
]
```

2. Implement the following operations:

- Save the student records to a CSV file named 'students.csv'
- Read the CSV file and add a new student
- Calculate and display the average grade
- Export students with grades above 80 to 'high_performers.csv'
- Convert the data to JSON format with proper indentation

2 Exercise 2: Database Operations

Objective

Create and manage a simple SQLite database using Python.

1. Create a library database with the following schema:

```
CREATE TABLE books (
    book_id INTEGER PRIMARY KEY,
    title TEXT NOT NULL,
    author TEXT NOT NULL,
    year INTEGER,
    available BOOLEAN DEFAULT TRUE
);
```

2. Implement the following functions:

- create_database(): Create the database and table
- add_book(title, author, year): Add a new book
- list_available_books(): Show all available books
- find_books_by_author(author): List all books by an author

3. Write a program that:

- Adds at least 5 sample books
- Performs a search operation
- Exports results to a DataFrame

3 Exercise 3: API Integration

Objective

Fetch and process data from The Movie Database (TMDB) API

get your own API Keys in https://www.themoviedb.org/settings/api/request

1. Using the endpoint: `https://api.themoviedb.org/3/movie/top_rated`

```
params = {
    'api_key': 'your_api_key',
    'language': 'en-US',
    'page': 1
}
```

2. Implement the following features:

- Fetch the top 20 rated movies
- Extract title, release date, vote average, and popularity
- Create a pandas DataFrame with the results
- Save the data to 'top_movies.csv'

3. Calculate and display:

- Average rating of all movies
- Highest and lowest rated movies
- Movies released in the last 5 years

4 Exercise 4: Data Analysis with Pandas

Objective

Analyze student performance data using pandas.

1. Create a Pandas DataFrame with the following structure:

```
student_data = {
    'name': ['John', 'Alice', 'Bob', 'Sarah', 'Mike'],
    'age': [20, 21, 19, 22, 20],
    'math_score': [85, 90, 75, 95, 80],
    'science_score': [75, 95, 80, 85, 90],
    'passed': [True, True, False, True, True]
}
```

2. Perform the following analyses:

- Calculate average score for each student
- Find students with average score above 85
- Sort students by average score in descending order
- Display basic statistics for each subject

3. Create visualizations showing:

- Score distribution for each subject
- Comparison of math vs science scores

5 Final Challenge

Objective

Create an integrated solution combining all previous concepts.

1. Create a sales analysis system with this data:

```
sales = {
    'date': ['2024-01-01', '2024-01-02', '2024-01-03'],
    'product': ['A', 'B', 'A'],
    'quantity': [10, 15, 12],
    'price': [100, 150, 100]
}
```

2. Implement the following features:

- Create a DataFrame from the sales data
- Calculate daily revenue
- Identify the best-selling product
- Create visualizations of sales trends
- Export results to an Excel file