Managing Renewable Energy and Green Technology Projects with Pandas: Series, DataFrames, and Basic Operations ¶

Objectives:

1.Series Creation: We will create a Pandas Series to represent renewable energy sources.

2.DataFrame Creation: We will create a Pandas DataFrame to organize and manage data on green technology projects.

3.Basic Pandas Operations: We will implement basic operations with Pandas, such as accessing columns, filtering data, adding new columns, and performing aggregation and grouping.

Step 1: Import Pandas and Create a Dataset

First, we'll need to import the necessary library and create a dataset for renewable energy sources and green technology projects.

In [1]:
```
import pandas as pd

# Sample renewable energy sources data
renewable_sources = ["Solar", "Wind", "Hydropower", "Geothermal", "Biomass"]

# Sample green technology project data (for DataFrame)
data = {
    "Project": ["Solar Farm A", "Wind Turbine X", "Hydropower Y", "Solar Roo
    "Technology": ["Solar", "Wind", "Hydropower", "Solar", "Geothermal"],
    "Capacity (MW)": [150, 300, 200, 50, 100],  # Megawatts
    "Cost (Million $)": [200, 400, 350, 100, 250],  # Project cost
    "Location": ["California", "Texas", "Washington", "Nevada", "Idaho"],
    "Completion Year": [2023, 2024, 2022, 2025, 2023]
}
```

1. Create a Series for Renewable Energy Sources

A Pandas Series is a one-dimensional array-like structure. We will create a Series to represent renewable energy sources.

In [2]:
```
# Create a Pandas Series for renewable energy sources
renewable_series = pd.Series(renewable_sources)

# Print the Series
print("Renewable Energy Sources:")
print(renewable_series)
```

```
Renewable Energy Sources:
0         Solar
1          Wind
2    Hydropower
3    Geothermal
4       Biomass
dtype: object
```

The Pandas Series represents the renewable energy sources (solar, wind, hydropower, etc.). The Series is a simple data structure used to store one-dimensional data.

2: Create a DataFrame for Green Technology Projects

Now, we'll create a Pandas DataFrame using the data dictionary that stores information about various green technology projects, such as project names, technologies, capacities, and costs.

In [4]:
```
# Create a DataFrame for green technology projects
projects_df = pd.DataFrame(data)

# Print the DataFrame
print("\nGreen Technology Projects DataFrame:")
print(projects_df)
```

```
Green Technology Projects DataFrame:
              Project  Technology  Capacity (MW)  Cost (Million $)  \
0        Solar Farm A       Solar            150               200
1      Wind Turbine X        Wind            300               400
2        Hydropower Y  Hydropower            200               350
3        Solar Roof Z       Solar             50               100
4  Geothermal Plant P  Geothermal            100               250

     Location  Completion Year
0  California             2023
1       Texas             2024
2  Washington             2022
3      Nevada             2025
4       Idaho             2023
```

The DataFrame is a two-dimensional table-like structure. It organizes our green technology project data, such as project name, technology used, capacity, cost, location, and completion year.

3. Basic Pandas Operations

3.1. Accessing Columns

We can access individual columns of the DataFrame to get specific project attributes.

In [5]:
```
# Access the 'Project' column
print("\nList of Projects:")
print(projects_df["Project"])
```

```
List of Projects:
0          Solar Farm A
1        Wind Turbine X
2          Hydropower Y
3          Solar Roof Z
4    Geothermal Plant P
Name: Project, dtype: object
```

This code accesses the Project column to list all the green technology project names.

3.2. Filtering Data

Let's filter the projects based on a certain condition. For example, we want to see projects that have a capacity greater than 100 MW.

In [6]:
```
# Filter projects with capacity greater than 100 MW
high_capacity_projects = projects_df[projects_df["Capacity (MW)"] > 100]

print("\nProjects with Capacity Greater than 100 MW:")
print(high_capacity_projects)
```

```
Projects with Capacity Greater than 100 MW:
          Project  Technology  Capacity (MW)  Cost (Million $)    Location
\
0    Solar Farm A       Solar            150               200  California
1  Wind Turbine X        Wind            300               400       Texas
2    Hydropower Y  Hydropower            200               350  Washington

   Completion Year
0             2023
1             2024
2             2022
```

This filter operation selects projects that have a capacity greater than 100 MW using a condition in Pandas.

3.3. Adding New Columns

We'll add a new column to calculate the cost per megawatt for each project.

In [7]:
```
# Add a new column for cost per MW
projects_df["Cost per MW"] = projects_df["Cost (Million $)"] / projects_df["

print("\nDataFrame with Cost per MW:")
print(projects_df)
```

```
DataFrame with Cost per MW:
              Project  Technology  Capacity (MW)  Cost (Million $)  \
0        Solar Farm A       Solar            150               200
1      Wind Turbine X        Wind            300               400
2        Hydropower Y  Hydropower            200               350
3        Solar Roof Z       Solar             50               100
4  Geothermal Plant P  Geothermal            100               250

     Location  Completion Year  Cost per MW
0  California             2023     1.333333
1       Texas             2024     1.333333
2  Washington             2022     1.750000
3      Nevada             2025     2.000000
4       Idaho             2023     2.500000
```

This code adds a new column Cost per MW by dividing the total project cost by its capacity in megawatts.

3.4. Aggregation

We can aggregate the data to find the total capacity and total cost of all projects.

In [8]:
```
# Aggregate the total capacity and cost
total_capacity = projects_df["Capacity (MW)"].sum()
total_cost = projects_df["Cost (Million $)"].sum()

print(f"\nTotal Capacity of all projects: {total_capacity} MW")
print(f"Total Cost of all projects: ${total_cost} million")
```

```
Total Capacity of all projects: 800 MW
Total Cost of all projects: $1300 million
```

The sum() function aggregates the total capacity and cost across all projects.

3.5. Grouping Data

Finally, let's group the data by technology type and calculate the total capacity for each type of technology.

In [9]:
```
# Group by 'Technology' and calculate total capacity for each type
grouped_data = projects_df.groupby("Technology")["Capacity (MW)"].sum()

print("\nTotal Capacity by Technology:")
print(grouped_data)
```

```
Total Capacity by Technology:
Technology
Geothermal    100
Hydropower    200
Solar         200
Wind          300
Name: Capacity (MW), dtype: int64
```

This code groups the data by the Technology column and calculates the total capacity for each type of renewable energy (e.g., solar, wind, hydropower).

Conclusion

In this lab assignment, we created a Pandas Series to represent renewable energy sources and a DataFrame to organize green technology project data. We performed essential operations with Pandas, including:

```
Accessing columns to extract specific data,
Filtering the data based on conditions,
Adding new columns to enhance the dataset with additional informati
on,
Aggregating data to calculate total capacity and cost, and
Grouping data by technology to analyze the total capacity for each
energy type.
```

These operations help manage and analyze our datasets efficiently, making Pandas an invaluable tool for working with structured data in Python.