Certainly! Let's dive deeper into some additional examples and usages of pandas.

### 1. Grouping and Aggregation:

```python

# Grouping data by a column and calculating aggregate statistics

grouped\_data = df.groupby('Gender').agg({'Age': ['mean', 'median', 'min', 'max']})

print(grouped\_data)

```

### 2. Merging DataFrames:

```python

# Creating two DataFrames

df1 = pd.DataFrame({'ID': [1, 2, 3], 'Name': ['Alice', 'Bob', 'Charlie']})

df2 = pd.DataFrame({'ID': [2, 3, 4], 'Age': [25, 30, 35]})

# Merging DataFrames based on a common column

merged\_df = pd.merge(df1, df2, on='ID', how='inner')

print(merged\_df)

```

### 3. Reshaping Data:

```python

# Pivoting data

pivot\_table = df.pivot\_table(index='Name', columns='Gender', values='Age', aggfunc='mean')

print(pivot\_table)

```

### 4. Applying Functions:

```python

# Applying a function to a column

df['Age'] = df['Age'].apply(lambda x: x \* 2)

print(df)

# Applying a function element-wise

df['City'] = df['City'].apply(lambda x: x.upper())

print(df)

```

### 5. Concatenating DataFrames:

```python

# Concatenating DataFrames

df1 = pd.DataFrame({'A': [1, 2, 3], 'B': [4, 5, 6]})

df2 = pd.DataFrame({'A': [4, 5, 6], 'B': [7, 8, 9]})

concatenated\_df = pd.concat([df1, df2], ignore\_index=True)

print(concatenated\_df)

```

### 6. Handling Time Series Data:

```python

# Creating a time series DataFrame

dates = pd.date\_range(start='2022-01-01', end='2022-01-05')

time\_series\_df = pd.DataFrame({'Date': dates, 'Value': [10, 20, 15, 25, 30]})

print(time\_series\_df)

# Resampling time series data

weekly\_mean = time\_series\_df.resample('W-Mon', on='Date').mean()

print(weekly\_mean)

```

### 7. Handling Categorical Data:

```python

# Creating a DataFrame with categorical data

data = {'Name': ['Alice', 'Bob', 'Charlie', 'David'],

'Gender': ['Female', 'Male', 'Male', 'Male']}

df = pd.DataFrame(data)

df['Gender'] = df['Gender'].astype('category')

print(df['Gender'].cat.categories) # Display categories

```

### 8. Handling JSON Data:

```python

# Reading JSON data into a DataFrame

json\_data = '{"Name": ["Alice", "Bob", "Charlie"], "Age": [25, 30, 35]}'

df = pd.read\_json(json\_data)

print(df)

```

### 9. Working with Excel Files:

```python

# Reading an Excel file into a DataFrame

excel\_df = pd.read\_excel('data.xlsx', sheet\_name='Sheet1')

print(excel\_df)

```

These examples cover a range of operations and use cases in pandas. Experiment with them to get a better understanding of how to leverage pandas for your data analysis needs.