

# Problem Set 1 - Reshaping and Method Chaining in Pandas

## 1. Pivoting

### Easy

You have a DataFrame containing daily sales data:

```
df = pd.DataFrame({
    'Date': ['2024-11-01', '2024-11-01', '2024-11-02',
    ↪ ],
    'Product': ['A', 'B', 'A'],
    'Sales': [100, 200, 150]
})
```

Pivot the DataFrame so that the rows are **Date** and the columns are **Product**, with values being the **Sales**.

### Medium

A DataFrame contains temperature data:

```
df = pd.DataFrame({
    'City': ['A', 'A', 'B', 'B'],
    'Date': ['2024-11-01', '2024-11-02', '2024-11-01',
    ↪ , '2024-11-02'],
    'Temperature': [20, 22, 25, 26],
    'Humidity': [60, 65, 70, 72]
})
```

Create a pivot where rows are **City**, columns are **Date**, and the values are **both Temperature and Humidity**.

## Hard

Using the same DataFrame, create a pivot table where rows are `City`, columns are `Date`, and the values are the **average of Temperature and Humidity**.

## 2. Melting

### Easy

You have the following pivoted DataFrame:

```
df = pd.DataFrame({
    '2024-11-01': [20, 25],
    '2024-11-02': [22, 26]
}, index=['A', 'B'])
df.index.name = 'City'
df.columns.name = 'Date'
```

Unpivot this DataFrame to long format with columns `Date`, `City`, and `Temperature`.

### Medium

Given a DataFrame with multiple categories:

```
df = pd.DataFrame({
    'City': ['A', 'B'],
    'Temp_2024-11-01': [20, 25],
    'Temp_2024-11-02': [22, 26],
    'Humidity_2024-11-01': [60, 70],
    'Humidity_2024-11-02': [65, 72]
})
```

Melt the DataFrame to have `City`, `Date`, `Measurement`, and `Value`.

### Hard

Using the melted DataFrame from the medium question, reshape it further by splitting the `Category` column into `Measurement` and `Date`.

### 3. Stacking/Unstacking

#### Easy

You have a pivoted DataFrame:

```
df = pd.DataFrame({
    '2024-11-01': [20, 25],
    '2024-11-02': [22, 26]
}, index=['A', 'B'])
df.index.name = 'City'
df.columns.name = 'Date'
```

Stack the DataFrame to move the Date column into the index.

#### Medium

Unstack the following DataFrame so that Category becomes a column:

```
df = pd.DataFrame({
    'Temperature': [20, 22, 25, 26],
    'Humidity': [60, 65, 70, 72]
}, index=pd.MultiIndex.from_product(
    [['A', 'B'], ['2024-11-01', '2024-11-02']],
    names=['City', 'Date']
))
```

#### Hard

Using the same DataFrame, stack and reset it to long format with columns City, Date, Measurement, and Value.

### 4. Method Chaining

#### Easy

Start with the following DataFrame:

```
df = pd.DataFrame({
    'City': ['A', 'B', 'A', 'B'],
    'Date': ['2024-11-01', '2024-11-01', '2024-11-02',
    ↪      '2024-11-02'],
```

```
'Sales': [100, 200, 150, 250]
})
```

Filter the DataFrame for rows where **Sales** > 150 and reset the index.

## Medium

From the same DataFrame, calculate the total **Sales** per **City** using method chaining.

## Hard

From the same DataFrame, calculate the total **Sales** per **City** and sort the results by **Sales** in descending order, all in one chain.