\*\*Task 1: Sorting a DataFrame\*\*

To sort a DataFrame by its index, you can use the `sort\_index` method. To sort by a specific column, you can use the `sort\_values` method.

```python

import pandas as pd

# Sorting by index

df\_sorted\_by\_index = df.sort\_index()

# Sorting by a specific column (e.g., 'column\_name')

df\_sorted\_by\_column = df.sort\_values(by='column\_name')

```

\*\*Task 2: Lambda Expression\*\*

A lambda expression (or lambda function) is a small anonymous function without a name. It can have any number of parameters but can only have one expression. They are typically used when you want to create a simple, one-line function that you don't want to define using the `def` keyword.

Example:

```python

# Lambda function to add two numbers

add = lambda x, y: x + y

result = add(3, 4) # Result will be 7

```

\*\*Task 3: Detect and Remove Outliers\*\*

To detect and remove outliers using pandas, you can use statistical methods like z-score or IQR (Interquartile Range). Here's an example using the z-score method:

```python

import pandas as pd

from scipy.stats import zscore

# Assuming 'df' is your DataFrame and 'column\_name' is the column you want to check for outliers

z\_scores = zscore(df['column\_name'])

outliers = (z\_scores > 3) | (z\_scores < -3)

# Remove outliers

df\_cleaned = df[~outliers]

```

\*\*Task 4: Read Random Rows\*\*

To read random rows from your data, you can use the `sample` method. For example, if you want to read 5 random rows:

```python

random\_rows = df.sample(n=5)

```

\*\*Task 5: Combine Multiple DataFrames\*\*

To combine multiple DataFrames, you can use methods like `concat`, `merge`, or `join` depending on how you want to combine them. Here's an example using `concat`:

```python

# Assuming df1, df2, df3 are your DataFrames

combined\_df = pd.concat([df1, df2, df3], axis=0) # Combine vertically (along rows)

```

\*\*Task 6: Replace Data in Specific Columns\*\*

You can use the `replace` method to replace specific values in specific columns.

```python

# Replace all occurrences of 'old\_value' with 'new\_value' in 'column\_name'

df['column\_name'] = df['column\_name'].replace('old\_value', 'new\_value')

```

\*\*Task 7: Write Dataframes to Separate Sheets in Excel\*\*

You can use the `to\_excel` method to write DataFrames to Excel. To write to separate sheets, you'll need to use the `ExcelWriter` class.

```python

with pd.ExcelWriter('output.xlsx', engine='xlsxwriter') as writer:

df1.to\_excel(writer, sheet\_name='Sheet1')

df2.to\_excel(writer, sheet\_name='Sheet2')

```

Make sure you have the `xlsxwriter` library installed (`pip install xlsxwriter`).