Certainly! Let's go through each of the topics: reading/writing data from/to various sources, and data transformations.

5. Data Sources:

Reading and Writing CSV Data:

Reading CSV:
```python
import pandas as pd

# Read CSV file into a DataFrame
csv_file_path = 'path/to/your/file.csv'
df = pd.read_csv(csv_file_path)

# Display the DataFrame
print(df.head())
```

Writing CSV:
```python
# Write DataFrame to CSV file
output_csv_path = 'path/to/your/output_file.csv'
df.to_csv(output_csv_path, index=False)
```

Reading and Writing JSON Data:

Reading JSON:
```python
# Read JSON file into a DataFrame
json_file_path = 'path/to/your/file.json'
df = pd.read_json(json_file_path)

# Display the DataFrame
print(df.head())
```

Writing JSON:
```python
# Write DataFrame to JSON file
output_json_path = 'path/to/your/output_file.json'
df.to_json(output_json_path, orient='records')
```

Reading and Writing Parquet Data:

Reading Parquet:
```python
import pyarrow.parquet as pq

# Read Parquet file into a DataFrame
parquet_file_path = 'path/to/your/file.parquet'
table = pq.read_table(parquet_file_path)
df = table.to_pandas()

# Display the DataFrame
print(df.head())
```

Writing Parquet:
```python
# Write DataFrame to Parquet file
output_parquet_path = 'path/to/your/output_file.parquet'
table = pa.Table.from_pandas(df)
pq.write_table(table, output_parquet_path)
```

6. Data Transformations:

Filtering, Sorting, and Aggregating Data:

Filtering:
```python
# Example: Filter rows where 'column_name' is greater than 10
filtered_df = df[df['column_name'] > 10]
```

Sorting:
```python
# Example: Sort DataFrame by 'column_name' in ascending order
sorted_df = df.sort_values(by='column_name')
```

Aggregating:
```python
# Example: Calculate the mean of a specific column
mean_value = df['column_name'].mean()
```

```

#### Joins and Unions:

**Joins:**
```python
# Example: Inner join two DataFrames on a common column
merged_df = pd.merge(df1, df2, on='common_column', how='inner')
```

Unions:
```python
# Example: Concatenate two DataFrames vertically (stacking them)
union_df = pd.concat([df1, df2], axis=0)
```
Certainly! Let's delve deeper into each aspect with additional examples:

5. Data Sources:

Reading and Writing CSV Data:

Reading CSV with Custom Parameters:
```python
# Read CSV file with custom parameters
custom_df = pd.read_csv(csv_file_path, delimiter=';', header=None, names=['col1', 'col2'])

# Display the DataFrame
print(custom_df.head())
```

Writing CSV with Specific Settings:
```python
# Write DataFrame to CSV file with specific settings
df.to_csv(output_csv_path, index=False, encoding='utf-8', sep='\t')
```

Reading and Writing JSON Data:

Reading JSON from API:
```python
import requests

# Read JSON from API into a DataFrame
api_url = 'https://api.example.com/data'
```
```

```
response = requests.get(api_url)
json_data = response.json()
api_df = pd.DataFrame(json_data)

# Display the DataFrame
print(api_df.head())
```

Writing JSON with Compact Format:
```python
# Write DataFrame to JSON file with compact format
df.to_json(output_json_path, orient='split', indent=0)
```

Reading and Writing Parquet Data:

Reading Parquet with Column Selection:
```python
# Read selected columns from Parquet file into a DataFrame
selected_columns = ['col1', 'col2']
table = pq.read_table(parquet_file_path, columns=selected_columns)
df_selected = table.to_pandas()

# Display the DataFrame
print(df_selected.head())
```

Writing Parquet with Compression:
```python
# Write DataFrame to Parquet file with compression
output_parquet_compressed_path = 'path/to/your/output_file_compressed.parquet'
table_compressed = pa.Table.from_pandas(df)
pq.write_table(table_compressed, output_parquet_compressed_path, compression='snappy')
```

6. Data Transformations:

Filtering, Sorting, and Aggregating Data:

Multiple Conditions Filtering:
```python
# Example: Filter rows where 'col1' is greater than 10 and 'col2' is 'A'
filtered_multi_condition = df[(df['col1'] > 10) & (df['col2'] == 'A')]
```

Descending Order Sorting:
```python
# Example: Sort DataFrame by 'col1' in descending order
sorted_desc_df = df.sort_values(by='col1', ascending=False)
```

Groupby and Aggregation:
```python
# Example: Group by 'col2' and calculate the mean for each group
grouped_df = df.groupby('col2')['col1'].mean().reset_index()
```

Joins and Unions:

Outer Join with Filling NaN:
```python
# Example: Outer join two DataFrames and fill NaN values with a specific value
merged_outer_df = pd.merge(df1, df2, on='common_column', how='outer').fillna('NA')
```

Union with Duplicate Removal:
```python
# Example: Concatenate two DataFrames vertically and remove duplicates
union_no_duplicates_df = pd.concat([df1, df2], ignore_index=True).drop_duplicates()
```

These additional examples showcase more advanced scenarios and flexibility in handling various data source formats and performing complex data transformations using pandas.