**✅ Part 1: Generalized Code Templates for Questions 1–32**

Each operation uses placeholder names like (column\_name), (dataset.csv), etc., so you can plug in your real dataset and column names later.

**🔁 General Setup (Import Libraries)**

import pandas as pd

import numpy as np

**📊 1. Subsetting Rows**

# Subset based on condition

df\_subset = df[(df['(column\_name1)'] > value) & (df['(column\_name2)'] == 'condition')]

**🔗 2. Merging Datasets**

# Merge with another dataset

merged\_df = pd.merge(df, df2, on='(common\_column)', how='inner') # Options: 'left', 'right', 'outer'

**📌 3. Sorting Data**

# Sort by multiple columns

df\_sorted = df.sort\_values(by=['(column\_name1)', '(column\_name2)'], ascending=[True, False])

**🔄 4. Transposing Data**

# Transpose first 5 rows

df.head().T

**🔁 5. Reshape Using melt() and pivot()**

# Melt to long format

df\_melted = pd.melt(df, id\_vars=['(id\_column)'], value\_vars=['(feature1)', '(feature2)'])

# Pivot back to wide format

df\_pivoted = df.pivot(index='(id\_column)', columns='(feature)', values='(value\_column)')

**🧹 6. Data Cleaning (missing values, duplicates, types)**

# Handling missing values

df.fillna(value=0, inplace=True)

df.dropna(inplace=True)

# Remove duplicates

df.drop\_duplicates(inplace=True)

# Check data types

print(df.dtypes)

**🔍 7. Column Operations**

# Creating new columns

df['(new\_column)'] = df['(column1)'] / df['(column2)']

# Converting data types

df['(column)'] = df['(column)'].astype(str)

**📏 8. Normalization Techniques**

# Min-Max

df['(normalized\_column)'] = (df['(column\_name)'] - df['(column\_name)'].min()) / (df['(column\_name)'].max() - df['(column\_name)'].min())

# Z-Score

df['(zscore\_column)'] = (df['(column\_name)'] - df['(column\_name)'].mean()) / df['(column\_name)'].std()

# Decimal Scaling

df['(decimal\_scaled)'] = df['(column\_name)'] / 10\*\*len(str(int(df['(column\_name)'].abs().max())))

**📈 9. Pivot Table / Groupby**

# Pivot table for aggregation

pivot = df.pivot\_table(index='(category\_column)', columns='(feature)', values='(value\_column)', aggfunc='mean')

# Group by example

grouped = df.groupby('(group\_column)').agg({'(value\_column)': ['mean', 'sum']})

**🧠 10. Label / One-Hot Encoding**

# Label Encoding

df['(column)'] = df['(column)'].astype('category').cat.codes

# One-Hot Encoding

df = pd.get\_dummies(df, columns=['(categorical\_column)'])

**🧼 11. Email Standardization & Error Correction**

# Standardize email addresses

df['(email\_column)'] = df['(email\_column)'].str.lower().str.replace(' ', '').str.replace('@.', '@')

# Error corrections

df.loc[df['(column\_name)'] > 120, '(column\_name)'] = 120