# Extract .7z files

import py7zr

import os

def extract\_7z(archive\_path, extract\_to, password=None):

with py7zr.SevenZipFile(archive\_path, mode='r', password=password) as archive:

archive.extractall(extract\_to)

# Replace 'your\_folder\_path' with the path to the folder containing .7z files

folder\_path = 'your\_folder\_path'

extract\_folder = 'extracted\_contents'

# Replace 'your\_password' with the actual password for the .7z files

password\_for\_7z = 'your\_password'

# Iterate through each .7z file in the folder and extract them

for filename in os.listdir(folder\_path):

if filename.endswith('.7z'):

full\_path = os.path.join(folder\_path, filename)

extract\_7z(full\_path, extract\_folder, password\_for\_7z)

print(f"Extracted: {filename}")

print("Extraction completed for all .7z files.")

# task 2:

import zipfile

import os

import pandas as pd

def extract\_zip(zip\_path, extract\_to):

with zipfile.ZipFile(zip\_path, 'r') as zip\_ref:

zip\_ref.extractall(extract\_to)

def explore\_zip\_contents(zip\_path, extract\_to, result=None):

if result is None:

result = []

extract\_zip(zip\_path, extract\_to)

# Get the current folder path relative to the extraction root

current\_folder = os.path.relpath(extract\_to, start=os.path.dirname(zip\_path))

# Iterate through the extracted files and folders

for root, dirs, files in os.walk(extract\_to):

for file in files:

file\_path = os.path.join(root, file)

# Add information to the result list

file\_name, file\_extension = os.path.splitext(file)

result.append((current\_folder, file\_path, file\_name, file\_extension))

# Check if the file is a zip file and explore its contents recursively

if file\_path.endswith('.zip'):

new\_extract\_to = os.path.join(extract\_to, file\_name)

explore\_zip\_contents(file\_path, new\_extract\_to, result)

# Replace 'your\_main\_folder\_path' with the path to the main folder containing nested zip folders

main\_folder\_path = 'your\_main\_folder\_path'

extract\_folder = 'extracted\_contents'

# Initialize an empty list to store the extracted information

result\_list = []

# Iterate through each zip file in the main folder and extract them

for filename in os.listdir(main\_folder\_path):

if filename.endswith('.zip'):

full\_path = os.path.join(main\_folder\_path, filename)

explore\_zip\_contents(full\_path, extract\_folder, result\_list)

print(f"Extracted: {filename}")

# Convert the list to a DataFrame

df = pd.DataFrame(result\_list, columns=['Folder Name', 'File Path', 'File Name', 'File Extension'])

# Print the DataFrame

print(df)

import pandas as pd

# Replace 'input\_file.csv' with the actual CSV file path

input\_file\_path = 'input\_file.csv'

# Replace 'output\_file.txt' with the desired output text file path

output\_file\_path = 'output\_file.txt'

# Load the CSV file into a DataFrame

df = pd.read\_csv(input\_file\_path)

# Save the DataFrame to a text file with pipe delimiter

df.to\_csv(output\_file\_path, sep='|', index=False)

print(f"The CSV file '{input\_file\_path}' has been converted to a pipe-delimited text file '{output\_file\_path}'.")

===============

import pandas as pd

# Create a sample DataFrame

data = {'Column1': ['123ABC', '456DEF', '789GHI']}

df = pd.DataFrame(data)

# Display the original DataFrame

print("Original DataFrame:")

print(df)

# Specify the number of characters to remove from the beginning

characters\_to\_remove = 3

# Remove specified number of characters from the beginning of each observation

df['Column1'] = df['Column1'].str[characters\_to\_remove:]

# Display the modified DataFrame

print("\nDataFrame after removing characters:")

print(df)

—------------------------

import pandas as pd

# Create a sample DataFrame

data = {'Column1': ['123ABC pagees', '456DEF', '789GHI']}

df = pd.DataFrame(data)

# Display the original DataFrame

print("Original DataFrame:")

print(df)

# Replace "pagees" with "Pages" in the entire DataFrame

df = df.replace(to\_replace="pagees", value="Pages", regex=True)

# Display the modified DataFrame

print("\nDataFrame after replacing 'pagees' with 'Pages':")

print(df)

=============

# Convert the DataFrame to a text file with "|" as the delimiter

df.to\_csv(output\_file\_path, sep='|', index=False)