# Import necessary libraries

import pandas as pd # For data manipulation and analysis

import numpy as np # For numerical operations

import os # For operating system related operations

import glob # For file path pattern matching

import matplotlib.pyplot as plt # For plotting

import seaborn as sns # For statistical data visualization

from scipy.signal import butter, filtfilt # For signal processing (not used in this code snippet)

from sklearn.preprocessing import MinMaxScaler, StandardScaler # For data scaling (not used in this code snippet)

# Define the path to the dataset

datasetPath = r"Path way"

# Get a list of all CSV files in the specified directory

listCsvFiles = glob.glob(os.path.join(datasetPath, "\*.csv"))

# Read each CSV file into a pandas DataFrame and store in a list

dataFrameList = [pd.read\_csv(file) for file in listCsvFiles]

# Concatenate all DataFrames in the list into a single DataFrame

dataFrame = pd.concat(dataFrameList, ignore\_index=True)

# Print information about the DataFrame (column names, data types, non-null counts)

print(dataFrame.info())

# Convert integer columns to int32 to reduce memory usage

for col in dataFrame.select\_dtypes(include=["int64"]).columns:

dataFrame[col] = dataFrame[col].astype("int32")

# Convert float columns to float32 to reduce memory usage

for col in dataFrame.select\_dtypes(include=["float64"]).columns:

dataFrame[col] = dataFrame[col].astype("float32")

# Convert object (string) columns to categorical data type for efficiency

for col in dataFrame.select\_dtypes(include=["object"]).columns:

dataFrame[col] = dataFrame[col].astype("category")

# Uncomment the following line to check the reduced memory usage after type conversions

#print(dataFrame.info())