

```

from google.oauth2 import service_account
from googleapiclient.discovery import build
import pandas as pd
import requests
from io import StringIO
from io import BytesIO # For handling Excel files

# Replace with the path to your service account credentials
SERVICE_ACCOUNT_FILE = "Put the service account key json path"
SCOPES = ['https://www.googleapis.com/auth/drive.readonly']

# Authenticate and create the service
credentials = service_account.Credentials.from_service_account_file(
    SERVICE_ACCOUNT_FILE, scopes=SCOPES)
service = build('drive', 'v3', credentials=credentials)

# Replace with your Google Drive folder ID
FOLDER_ID = 'Put your Folder ID'

def list_files(service, folder_id):
    results = service.files().list(
        q=f'"{folder_id}" in parents',
        fields="files(id, name, mimeType)"
    ).execute()
    return results.get('files', [])

# Fetch the files
files = list_files(service, FOLDER_ID)

# List to store dataframes for file content
file_dataframes = []

for file in files:
    file_id = file['id']
    file_name = file['name']
    mime_type = file['mimeType']

    # Check if the file is Google Sheets, CSV, or Excel
    if mime_type == 'application/vnd.google-apps.spreadsheet':
        # Export Google Sheets as CSV
        download_url = f"https://docs.google.com/spreadsheets/d/{file_id}/export?format=csv"
    elif mime_type == 'text/csv':
        # If it's already a CSV file

```

```

        download_url = f"https://drive.google.com/uc?export=download&id={file_id}"
    elif mime_type == 'application/vnd.openxmlformats-officedocument.spreadsheetml.sheet':
        # If it's an Excel file
        download_url = f"https://drive.google.com/uc?export=download&id={file_id}"
    else:
        continue # Skip files that are not CSV, Google Sheets, or Excel

# Download and read the CSV or Excel file into a pandas dataframe
response = requests.get(download_url)
if response.status_code == 200:
    if mime_type == 'application/vnd.openxmlformats-officedocument.spreadsheetml.sheet':
        # For Excel, use BytesIO to read the binary content
        df = pd.read_excel(BytesIO(response.content))
    else:
        # For CSV and Google Sheets (converted to CSV)
        df = pd.read_csv(StringIO(response.content.decode('utf-8')))
    file_dataframes.append(df)
else:
    print(f"Error downloading file: {file_name}")

# Combine all dataframes into a single one
if file_dataframes:
    combined_df = pd.concat(file_dataframes, ignore_index=True)

```