pip install google-auth google-auth-oauthlib google-auth-httplib2 google-api-python-client

pip install matplotlib

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 = "C:\\Users\\Karan\\Downloads\\Financial Project in Power BI\\financial-project-drive-bridge-87e32fc7d02b.json"

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 = '1IFmd9klIO4qfG6EXFjmiKMAueWR31v2w'

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)