# 📘 Databricks Excel File Analysis with PySpark (Serverless Edition)

## 📁 Part 1: Upload the Excel File into DBFS

1. Go to Databricks Workspace.

2. Open your Notebook.

3. Click the sidebar 📁 'Data' > 'Add Data' > Upload File.

4. Select your Excel file: Financial Sample (2).xlsx

5. Target path: /FileStore/tables/financial\_sample.xlsx

## 📦 Part 2: Install and Read Excel File

Install openpyxl:

%pip install openpyxl

Read Excel using Pandas:

import pandas as pd  
excel\_path = "/dbfs/FileStore/tables/financial\_sample.xlsx"  
pdf = pd.read\_excel(excel\_path, engine="openpyxl")  
pdf.head()

## 🔄 Part 3: Convert to Spark DataFrame

df = spark.createDataFrame(pdf)  
df.show(5)

## 🔧 Part 4: Data Transformation Operations

1. View Schema

df.printSchema()

2. Filter Rows (Sales > 5000)

df.filter(df['Sales'] > 5000).show()

3. Replace Nulls

df = df.fillna({'Sales': 0, 'Country': 'Unknown'})

4. Drop Columns

df = df.drop('Segment')

5. Add New Column (Profit Margin)

from pyspark.sql.functions import col, round  
df = df.withColumn("Profit\_Margin", round(col("Profit") / col("Sales"), 2))

6. Change Data Type

df = df.withColumn('Sales', col('Sales').cast('double'))

7. Remove Duplicates

df = df.dropDuplicates()

8. Group and Aggregate

df.groupBy('Country').agg({'Sales': 'sum', 'Profit': 'sum'}).show()

## 📊 Part 5: Generate Business Reports

Report 1: Top 5 Countries by Sales

df.groupBy("Country") \  
 .sum("Sales") \  
 .orderBy("sum(Sales)", ascending=False) \  
 .show(5)

Report 2: Average Profit by Segment

df.groupBy('Segment').avg('Profit').show()

Report 3: Sales Trend by Month

from pyspark.sql.functions import month, year  
df = df.withColumn("Month", month(col("Date")))  
df.groupBy("Month").sum("Sales").orderBy("Month").show()

## ✅ Summary of Capabilities

- Read Excel with Pandas and convert to Spark  
- Clean null values using fillna()  
- Drop and add columns  
- Filter rows  
- Convert data types  
- Aggregate with groupBy() and agg()  
- Remove duplicates  
- Generate business reports using Spark transformations