**Migrating a More Complex SAS Example to PySpark**

Let's consider a more complex SAS example involving data cleaning, transformation, and aggregation:

**SAS Code:**

SAS

data clean\_data;

input id age gender $ income;

datalines;

1 25 M 50000

2 30 F 60000

3 28 M 45000

4 32 F 70000

5 27 M 55000

;

run;

proc means data=clean\_data mean stddev min max;

var income;

class gender;

run;

**PySpark Code:**

Python

from pyspark.sql import SparkSession

spark = SparkSession.builder.appName("SAS\_to\_PySpark").getOrCreate()

data = [(1, 25, "M", 50000),

(2, 30, "F", 60000),

(3, 28, "M", 45000),

(4, 32, "F", 70000),

(5, 27, "M", 55000)]

df = spark.createDataFrame(data, ["id", "age", "gender", "income"])

# Data Cleaning (if needed)

# ... (e.g., handling missing values, outliers)

# Aggregation

df.groupBy("gender").agg(

{"income": "mean"},

{"income": "stddev"},

{"income": "min"},

{"income": "max"}

).show()

**Explanation:**

1. **Data Creation:** We create a PySpark DataFrame from the given data.
2. **Data Cleaning:** While not explicitly shown in this example, you can perform data cleaning operations like handling missing values, outliers, and inconsistencies using PySpark's built-in functions or custom UDFs.
3. **Aggregation:** We use groupBy and agg to calculate the mean, standard deviation, minimum, and maximum income for each gender.