Case Study 2 (Python)

Import notebook

```
# Load dataset from S3
df = spark.read.option("sep", ",") \
    .option("header", True) \
    .option("inferSchema", True) \
    .csv("s3n://humber-lfb-databricks-class-files/midterm_la.csv")

# Show the first 10 rows
df.show(10)

# Print schema to understand structure
df.printSchema()
```

df: pyspark.sql.dataframe.DataFrame = [Emp ID: integer, Name Prefix: string ... 37 more fields]

```
|-- Month Name of Joining: string (nullable = true)
|-- Short Month: string (nullable = true)
|-- Day of Joining: integer (nullable = true)
|-- DOW of Joining: string (nullable = true)
|-- Short DOW: string (nullable = true)
|-- Age in Company (Years): double (nullable = true)
|-- Salary: integer (nullable = true)
|-- Last % Hike: string (nullable = true)
|-- SSN: string (nullable = true)
|-- Phone No. : string (nullable = true)
|-- Place Name: string (nullable = true)
|-- County: string (nullable = true)
|-- City: string (nullable = true)
|-- State: string (nullable = true)
|-- Zip: integer (nullable = true)
|-- Region: string (nullable = true)
|-- User Name: string (nullable = true)
|-- Password: string (nullable = true)
|-- _c37: string (nullable = true)
|-- CompanyID: integer (nullable = true)
```

Dataset Explanation

The dataset contains employee information such as demographics, salary, job position, and company tenure. Each row represents an individual employee, and the columns capture attributes relevant to HR analytics.

Key Columns:

Employee ID – Unique identifier for employees

Gender – Male, Female, or other

Region – Geographic region where employee works

Age – Employee age

Salary – Employee salary

Date of Joining – The date the employee joined the company

Age in Company (Years) – Tenure calculated from hire date

Job Role / Department – Position held by the employee

Dataset Structure

;

The schema shows column names, data types, and nullability.

Most fields are string, integer, or double, while Date of Joining is a date type.

The dataset is suitable for analyzing salary trends, employee demographics, and tenure.

```
# Convert Date of Joining to date type if necessary
from pyspark.sql.functions import col, to_date

df = df.withColumn("Date of Joining", to_date(col("Date of Joining"), "yyyy-MM-dd"))

# Select fields for time-series trend
trend_df = df.select("Date of Joining", "Age in Company (Years)") \
             .orderBy("Date of Joining")

display(trend_df)
```

```
df:  pyspark.sql.dataframe.DataFrame = [Emp ID: integer, Name Prefix: string ... 37 more fields]
trend_df:  pyspark.sql.dataframe.DataFrame = [Date of Joining: date, Age in Company (Years): double]
```

Table Visualization 1

Employees who joined earlier naturally show higher tenure.

Hiring surges may appear if many employees share the same start dates.

Recent joiners show lower tenures, indicating ongoing recruitment.

```
from pyspark.sql.functions import avg

avg_salary_region = df.groupBy("Region") \
                      .agg(avg("Salary").alias("avg_salary")) \
                      .orderBy("avg_salary", ascending=False)

display(avg_salary_region)
```

avg_salary_region: pyspark.sql.dataframe.DataFrame = [Region: string, avg_salary: double]

Table Visualization 1

```
df.createOrReplaceTempView("employees")
```

```
%sql
SELECT
    Gender,
    `Emp ID`,
    `First Name`,
    Salary
FROM employees
QUALIFY ROW_NUMBER() OVER (PARTITION BY Gender ORDER BY Salary DESC) <= 3
```

_sqldf: pyspark.sql.dataframe.DataFrame = [Gender: string, Emp ID: integer ... 2 more fields]

Table

This result is stored as `_sqldf` and can be used in other Python and SQL cells.

Insight 1 — Salary Variation by Region

Some regions show significantly higher average salaries, suggesting differences in cost of living, talent availability, or business function concentration.

Insight 2 — Gender Pay Differences

From the top-3 salary ranking, certain genders may dominate the highest-paying roles—indicating possible pay gaps or seniority imbalances.

Insight 3 — Hiring Trend Over Time

The line chart may reveal:

periods of heavy hiring,

steady growth,

or slow recruitment phases. This helps HR understand workforce expansion patterns.