# Informe: Operaciones Estadísticas con Apache Spark

## Introducción a Apache Spark

Apache Spark es una herramienta de procesamiento distribuido para Big Data que permite realizar análisis de grandes volúmenes de datos a gran velocidad. Sus características incluyen soporte para múltiples lenguajes, ejecución en memoria, y la capacidad de operar en clústeres, lo cual lo convierte en una opción popular para aplicaciones de análisis masivo de datos.

## Operaciones Estadísticas en Apache Spark

Apache Spark incluye funciones que permiten realizar análisis estadísticos para entender tendencias y relaciones dentro de conjuntos de datos. Las operaciones más comunes incluyen:

### 1. Media

La media se utiliza para encontrar el promedio de un conjunto de datos y se calcula sumando todos los valores y dividiéndolos por el total de elementos.

Ejemplo en Apache Spark: Para un conjunto de salarios de empleados almacenado en un DataFrame, usa el método `mean()` para calcular el salario promedio:  
  
```python  
from pyspark.sql import SparkSession  
spark = SparkSession.builder.appName('Estadistica').getOrCreate()  
data = [(1, 3500), (2, 4200), (3, 5000)]  
df = spark.createDataFrame(data, ['ID', 'Salario'])  
df.selectExpr('avg(Salario) as Salario\_Promedio').show()  
```

### 2. Mediana

La mediana es el valor central en un conjunto de datos ordenados, útil para entender el centro de la distribución sin influencia de valores atípicos extremos.

Ejemplo en Apache Spark: Para calcular la mediana en un DataFrame de Spark, usa el método `approxQuantile()`:  
  
```python  
percentiles = df.approxQuantile('Salario', [0.5], 0.01)  
print(f'Mediana: {percentiles[0]}')  
```

### 3. Moda

La moda es el valor más frecuente en el conjunto de datos, lo cual ayuda a identificar patrones o tendencias dominantes.

Ejemplo en Apache Spark: Aunque Spark no tiene un método directo para moda, puedes agrupar y contar los datos, seleccionando el valor con la mayor frecuencia:  
  
```python  
moda\_df = df.groupBy('Salario').count().orderBy('count', ascending=False).first()  
print(f'Moda: {moda\_df[0]}')  
```

### 4. Correlación

La correlación analiza la relación entre dos variables. Un coeficiente de correlación cercano a 1 indica una relación positiva fuerte, mientras que uno cercano a -1 indica una relación negativa fuerte. Un valor cerca de 0 sugiere poca o ninguna relación.

### 1. Correlación de Pearson

Descripción: La correlación de Pearson mide la relación lineal entre dos variables cuantitativas.

Ejemplo en Apache Spark: Calcula la correlación de Pearson entre dos columnas de un DataFrame:  
  
```python  
df.stat.corr('Salario', 'Experiencia', 'pearson')  
```

### 2. Correlación de Spearman

Descripción: La correlación de Spearman mide relaciones monótonas, útil para datos ordinales o no lineales.

Ejemplo en Apache Spark: Calcula la correlación de Spearman:  
  
```python  
df.stat.corr('Satisfaccion', 'Tiempo\_En\_Empresa', 'spearman')  
```

## Conclusión

Apache Spark permite realizar tanto análisis de correlación de Pearson como de Spearman, lo que proporciona flexibilidad para distintos tipos de datos y relaciones. Para datos continuos y distribuciones normales, Pearson es la opción más adecuada. Para datos ordinales, relaciones no lineales o datos con valores atípicos, Spearman es una mejor elección.