Paso 1) Funcionamiento básico con PySpark

En primer lugar, debe inicializar el SQLContext aún no está iniciado.

```
#from pyspark.sql import SQLContext
url = "https://raw.githubusercontent.com/guru99-edu/R-Programming/master/adult_data.csv"
from pyspark import SparkFiles
sc.addFile(url)
sqlContext = SQLContext(sc)
```

luego, puede leer el archivo cvs con SQLContext.read.csv. Utilice InferSchema establecido en True para indicar a Spark que adivine automáticamente el tipo de datos. De forma predeterminada, se convierte en False.

```
df = sqlContext.read.csv(SparkFiles.get("adult_data.csv"), header=True, inferSchema=
True)
```

Echemos un vistazo al tipo de datos

```
df.printSchema()
root
 |-- age: integer (nullable = true)
 |-- workclass: string (nullable = true)
 |-- fnlwgt: integer (nullable = true)
 |-- education: string (nullable = true)
 |-- education_num: integer (nullable = true)
 |-- marital: string (nullable = true)
 |-- occupation: string (nullable = true)
 |-- relationship: string (nullable = true)
 |-- race: string (nullable = true)
 |-- sex: string (nullable = true)
 |-- capital_gain: integer (nullable = true)
 |-- capital_loss: integer (nullable = true)
 |-- hours_week: integer (nullable = true)
 |-- native_country: string (nullable = true)
 |-- label: string (nullable = true)
```

Puedes ver los datos con show.

```
df.show(5, truncate = False)
```

```
+---+----------------+------+---------+-------------+------------------+----------------
-+-------------+-----+------+------------+------------+----------+--------------+-----+
|age|workclass       |fnlwgt|education|education_num|marital           |occupation
|relationship |race |sex   |capital_gain|capital_loss|hours_week|native_country|label|
+---+----------------+------+---------+-------------+------------------+----------------
-+-------------+-----+------+------------+------------+----------+--------------+-----+
|39 |State-gov       |77516 |Bachelors|13           |Never-married     |Adm-clerical
|Not-in-family|White|Male  |2174        |0           |40        |United-States |<=50K|
|50 |Self-emp-not-inc|83311 |Bachelors|13           |Married-civ-spouse|Exec-managerial
|Husband      |White|Male  |0           |0           |13        |United-States |<=50K|
|38 |Private         |215646|HS-grad  |9            |Divorced          |Handlers-
cleaners|Not-in-family|White|Male  |0           |0           |40        |United-States
|<=50K|
|53 |Private         |234721|11th     |7            |Married-civ-spouse|Handlers-
cleaners|Husband      |Black|Male  |0           |0           |40        |United-States
|<=50K|
|28 |Private         |338409|Bachelors|13           |Married-civ-spouse|Prof-specialty
|Wife         |Black|Female|0           |0           |40        |Cuba          |<=50K|
+---+----------------+------+---------+-------------+------------------+----------------
-+-------------+-----+------+------------+------------+----------+--------------+-----+
only showing top 5 rows
```

Si no configuró IndersHema en True, esto es lo que está sucediendo con el tipo. Hay todos en cadena.

```
df_string = sqlContext.read.csv(SparkFiles.get("adult.csv"), header=True, inferSchema=
False)
df_string.printSchema()
root
 |-- age: string (nullable = true)
 |-- workclass: string (nullable = true)
 |-- fnlwgt: string (nullable = true)
 |-- education: string (nullable = true)
 |-- education_num: string (nullable = true)
 |-- marital: string (nullable = true)
 |-- occupation: string (nullable = true)
 |-- relationship: string (nullable = true)
 |-- race: string (nullable = true)
 |-- sex: string (nullable = true)
 |-- capital_gain: string (nullable = true)
 |-- capital_loss: string (nullable = true)
 |-- hours_week: string (nullable = true)
 |-- native_country: string (nullable = true)
 |-- label: string (nullable = true)
```

Para convertir la variable continua en el formato correcto, puede usar la refundición de las columnas. Puede usar WithColumn para indicar a Spark qué columna operar la transformación.

```
# Import all from `sql.types`
from pyspark.sql.types import *

# Write a custom function to convert the data type of DataFrame columns
def convertColumn(df, names, newType):
    for name in names:
        df = df.withColumn(name, df[name].cast(newType))
    return df
# List of continuous features
CONTI_FEATURES  = ['age', 'fnlwgt','capital_gain', 'education_num', 'capital_loss',
'hours_week']
# Convert the type
df_string = convertColumn(df_string, CONTI_FEATURES, FloatType())
# Check the dataset
df_string.printSchema()
root
 |-- age: float (nullable = true)
 |-- workclass: string (nullable = true)
 |-- fnlwgt: float (nullable = true)
 |-- education: string (nullable = true)
 |-- education_num: float (nullable = true)
 |-- marital: string (nullable = true)
 |-- occupation: string (nullable = true)
 |-- relationship: string (nullable = true)
 |-- race: string (nullable = true)
 |-- sex: string (nullable = true)
 |-- capital_gain: float (nullable = true)
 |-- capital_loss: float (nullable = true)
 |-- hours_week: float (nullable = true)
 |-- native_country: string (nullable = true)
 |-- label: string (nullable = true)
```

```
from pyspark.ml.feature import StringIndexer
#stringIndexer = StringIndexer(inputCol="label", outputCol="newlabel")
#model = stringIndexer.fit(df)
#df = model.transform(df)
df.printSchema()
```

Seleccionar columnas

Puede seleccionar y mostrar las filas con selección y los nombres de las entidades. A continuación, se seleccionan edad y fnlwgt.

```
df.select('age','fnlwgt').show(5)
```

```
+---+------+
|age|fnlwgt|
+---+------+
| 39| 77516|
| 50| 83311|
| 38|215646|
| 53|234721|
| 28|338409|
+---+------+
only showing top 5 rows
```

Recuento por grupo

Si desea contar el número de ocurrencia por grupo, puede encadenar:

- GrupoPor ()
- count ()

juntos. En el ejemplo siguiente, cuenta el número de filas por nivel educativo.

```
df.groupBy("education").count().sort("count",ascending=True).show()
```

```
+------------+-----+
|   education|count|
+------------+-----+
|   Preschool|   51|
|     1st-4th|  168|
|     5th-6th|  333|
|   Doctorate|  413|
|        12th|  433|
|         9th|  514|
| Prof-school|  576|
|     7th-8th|  646|
|        10th|  933|
|  Assoc-acdm| 1067|
|        11th| 1175|
|   Assoc-voc| 1382|
|     Masters| 1723|
|   Bachelors| 5355|
|Some-college| 7291|
|     HS-grad|10501|
+------------+-----+
```

Describir los datos

Para obtener un resumen de estadísticas, de los datos, puede utilizar describe (). Se calculará el:

- conteo
- significar
- desviación estándar
- mín
- Máx

```
df.describe().show()
```

```
+-------+------------------+-----------+------------------+------------+----------------
-+--------+----------------+------------+------------------+------+-----------------+--
--------------+------------------+--------------+-----+
|summary|               age|  workclass|            fnlwgt|   education|
education_num| marital|      occupation|relationship|              race|   sex|
capital_gain|    capital_loss|        hours_week|native_country|label|
+-------+------------------+-----------+------------------+------------+----------------
-+--------+----------------+------------+------------------+------+-----------------+--
--------------+------------------+--------------+-----+
|  count|             32561|      32561|             32561|       32561|
32561|   32561|           32561|       32561|             32561| 32561|
32561|           32561|             32561|         32561|32561|
|   mean| 38.58164675532078|       null|189778.36651208502|        null|
10.0806793403151|    null|            null|        null|              null|
null|1077.6488437087312| 87.303829734959|40.437455852092995|          null| null|
| stddev|13.640432553581356|       null|105549.97769702227|
null|2.572720332067397|    null|            null|        null|              null|  null|
7385.292084840354|402.960218649002|12.347428681731838|          null| null|
|    min|               17|          ?|             12285|        10th|
1|Divorced|              ?|     Husband|Amer-Indian-Eskimo|Female|                0|
```

Si desea que la estadística de resumen de una sola columna, agregue el nombre de la columna dentro de describe ()

```
df.describe('capital_gain').show()
```

```
+-------+------------------+
|summary|      capital_gain|
+-------+------------------+
|  count|             32561|
|   mean|1077.6488437087312|
| stddev| 7385.292084840354|
|    min|                 0|
|    max|             99999|
+-------+------------------+
```

Filtrar datos

Puede utilizar filter () para aplicar estadísticas descriptivas en un subconjunto de datos. Por ejemplo, puede contar el número de personas mayores de 40 años

```
df.filter(df.age > 40).count()
```

13443

Estadísticas descriptivas por grupo

Finalmente, puede agrupar datos por grupo y calcular operaciones estadísticas como la media.

```
df.groupby('marital').agg({'capital_gain': 'mean'}).show()
```

```
+--------------------+------------------+
|             marital| avg(capital_gain)|
+--------------------+------------------+
|           Separated| 535.5687804878049|
|       Never-married|376.58831788823363|
|Married-spouse-ab...| 653.9832535885167|
|            Divorced| 728.4148098131893|
|             Widowed| 571.0715005035247|
|   Married-AF-spouse| 432.6521739130435|
|  Married-civ-spouse|1764.8595085470085|
+--------------------+------------------+
```