Instituto Tecnológico de Tijuana

Nombre de Facultad

Ingeniería Informática

Proyecto / Tarea / Practica:

Examen U1

Materia:

Datos Masivos

Facilitador:

Jose Christian

Alumnos:

Erik Saul Rivera Reyes

Brayan Baltazar Moreno

Fecha:

Tijuana Baja California a 21 de 03 2022

Codigo

```
import org.apache.spark.sql.SparkSession
var spark = SparkSession.builder().getOrCreate()


var df = spark.read.option("header",
"true").option("inferSchema","true")csv("G:/COSAS
PC/FILES/ESCUELA/DECIMO/DATOS_MASIVOS/PRACTICAS/Netflix_2011_2016.csv")
df.show()

//#3 MUESTRAS LOS NOMBRES DE LAS COLUMNAS
df.columns

//#4 MUESTRA EL ESQUEMA
df.printSchema()

//#5 IMPRIME LAS PRIMERAS 5 COLUMNAS
df.select($"Date", $"Open", $"High", $"Low", $"Close").show(5)
//IMPRIME TODAS LAS COLUMNAS
df.select($"Date", $"Open", $"High", $"Low", $"Close").show()

//#6 DESCRIBE EL ARCHIVO CSV QUE SE ESTA UTILIZANDO
df.describe().show()

//#7 CREANDO HV RATIO
var hvratio = df.withColumn("HV Ratio",df("High")/df("Volume"))
hvratio.show()

//#8 PICO MAS ALTO DE LA COLUMNA OPEN
df.select($"Date", $"Open").show(1)

//#9 SIGNIFICADO DE CLOSE CON TUS PALABRAS
//NOS EXPLICA COMO NETFLIX TERMINA SUS CUENTAS TENIENDO EN CUENTA TANTO
SUS ALTOS COMO BAJOS EN BASE A LOS DATOS PROPORCIONADOS

//#10 MAXIMO Y MINIMO DE LA COLUMNA VOLUMEN
df.select(max ($"Volume")).show()
df.select(min ($"Volume")).show()

//#11 Con Sintaxis Scala/Spark $ conteste los siguiente:


//A CUANTOS DIAS FUE LA COLUMNA CLOSE INFERIOR A 600?
var a = df.filter($"Close" < 600).count()


//B QUE PORCENTAJE DEL TIEMPO FUE LA COLUMNA HIGH MAYOR A 500
var b = (df.filter($"High" > 500).count()*100)/1260
```

```
//C ¿Cuál es la correlación de Pearson entre columna “High” y la columna “Volumen”?
df.select(corr("High", "Volume").alias("correlacion de Pearson")).show()

//D ¿Cuál es el máximo de la columna “High” por año?
df.groupBy(year(df("Date")).alias("Year")).max("High").sort(asc("Year")).show()


//E ¿Cuál es el promedio de columna “Close” para cada mes del calendario?
df.groupBy(month(df("Date")).alias("Month")).avg("Close").sort(asc("Month")).show()
```

Capturas de pantalla del examen

3)

```
scala> df.columns
res1: Array[String] = Array(Date, Open, High, Low, Close, Volume, Adj Close)
```

4)

```
scala> df.printSchema()
root
 |-- Date: timestamp (nullable = true)
 |-- Open: double (nullable = true)
 |-- High: double (nullable = true)
 |-- Low: double (nullable = true)
 |-- Close: double (nullable = true)
 |-- Volume: integer (nullable = true)
 |-- Adj Close: double (nullable = true)
```

5)

```
scala> df.select($"Date", $"Open", $"High", $"Low", $"Close").show(5)
+-------------------+----------+------------------+----------+-----------------+
|               Date|      Open|              High|       Low|            Close|
+-------------------+----------+------------------+----------+-----------------+
|2011-10-24 00:00:00|119.100002|120.28000300000001|115.100004|       118.839996|
|2011-10-25 00:00:00| 74.899999|         79.390001| 74.249997|        77.370002|
|2011-10-26 00:00:00|     78.73|         81.420001| 75.399997|        79.400002|
|2011-10-27 00:00:00| 82.179998| 82.71999699999999| 79.249998|80.86000200000001|
|2011-10-28 00:00:00| 80.280002|         84.660002| 79.599999|84.14000300000001|
+-------------------+----------+------------------+----------+-----------------+
only showing top 5 rows
```

6)

```
scala> df.describe().show()
+-------+------------------+------------------+------------------+------------------+--------------------+------------------+
|summary|              Open|              High|               Low|             Close|              Volume|         Adj Close|
+-------+------------------+------------------+------------------+------------------+--------------------+------------------+
|  count|              1259|              1259|              1259|              1259|                1259|              1259|
|   mean|230.39351086656092|233.97320872915006|226.80127876251044| 230.522453845909|2.5634836060365368E7|55.610540036536875|
| stddev|164.37456353264244| 165.9705082667129| 162.6506358235739|164.40918905512854| 2.306312683388607E7|35.186669331525486|
|    min|         53.990001|         55.480001|             52.81|              53.8|             3531300|             7.685714|
|    max|        708.900017|        716.159996|        697.569984|        707.610001|           315541800|        130.929993|
+-------+------------------+------------------+------------------+------------------+--------------------+------------------+
```

7)

```
scala> var hvratio = df.withColumn("HV Ratio",df("High")/df("Volume"))
hvratio: org.apache.spark.sql.DataFrame = [Date: timestamp, Open: double ... 6 more fields]

scala> hvratio.show()
+-------------------+-----------------+-----------------+---------+-----------------+--------+-----------------+--------------------+
|               Date|             Open|             High|      Low|            Close|  Volume|        Adj Close|            HV Ratio|
+-------------------+-----------------+-----------------+---------+-----------------+--------+-----------------+--------------------+
|2011-10-24 00:00:00|       119.100002|120.28000300000001|115.100004|       118.839996|120460200|        16.977142|9.985040951285156E-7|
|2011-10-25 00:00:00|        74.899999|        79.390001| 74.249997|        77.370002|315541800|11.052857000000001|2.515989989281927E-7|
|2011-10-26 00:00:00|            78.73|        81.420001| 75.399997|        79.400002|148733900|        11.342857|5.474206014903126E-7|
|2011-10-27 00:00:00|        82.179998| 82.71999699999999| 79.249998|80.86000200000001| 71190000|11.551428999999999|1.161960907430818...|
|2011-10-28 00:00:00|        80.280002|        84.660002| 79.599999|84.14000300000001| 57769600|            12.02|1.465476686700271...|
|2011-10-31 00:00:00|83.63999799999999|        84.090002| 81.450002|        82.080003| 39653600|        11.725715|2.120614572195210...|
|2011-11-01 00:00:00|        80.109998|        80.999998|     78.74|        80.089997| 33016200|        11.441428|2.453341026526372E-6|
|2011-11-02 00:00:00|        80.709998|        84.400002| 80.109998|        83.389999| 41384000|        11.912857|2.039435578967717E-6|
|2011-11-03 00:00:00|        84.130003|        92.600003| 81.800003|        92.290003| 94685500|13.184285999999998| 9.779744839494496E-7|
|2011-11-04 00:00:00|91.46999699999999| 92.89000300000001| 87.749999|        90.019998| 84483700|            12.86|1.099502069629999...|
|2011-11-07 00:00:00|             91.0|        93.839998| 89.979997|        90.830003| 47485200|        12.975715|1.976194645910725...|
|2011-11-08 00:00:00|91.22999899999999|        92.600003| 89.650002|        90.470001| 31906000|        12.924286|2.902275528113834...|
|2011-11-09 00:00:00|        89.000001|        90.440001| 87.999998|        88.049999| 28756000|        12.578571|3.145082800111281E-6|
|2011-11-10 00:00:00|        89.290001| 90.29999699999999| 84.839999|85.11999899999999| 39614400|            12.16|2.279474054889131E-6|
|2011-11-11 00:00:00|        85.899997|        87.949997|      83.7|        87.749999| 38140200|        12.535714|2.305965805108520...|
|2011-11-14 00:00:00|        87.989998|             88.1|     85.45|        85.719999| 21811300|        12.245714|4.039190694731629...|
|2011-11-15 00:00:00|             85.15|        87.050003| 84.499998|        86.279999| 21372400|        12.325714|4.073010190713256...|
|2011-11-16 00:00:00|        86.460003|        86.460003| 80.890002|        81.180002| 34560400|11.597142999999999|2.501707242971725E-6|
|2011-11-17 00:00:00|             80.77|        80.999998| 75.789999|        76.460001| 52823400|        10.922857|1.533411291208063...|
|2011-11-18 00:00:00|             76.7|        78.999999| 76.039998|        78.059998| 34729100|        11.151428|2.274749388841058...|
+-------------------+-----------------+-----------------+---------+-----------------+--------+-----------------+--------------------+
only showing top 20 rows
```

8)

```
scala> df.select($"Date", $"Open").show(1)
+-------------------+----------+
|               Date|      Open|
+-------------------+----------+
|2011-10-24 00:00:00|119.100002|
+-------------------+----------+
only showing top 1 row
```

9)

10)

```
scala> df.select(max ($"Volume")).show()
+-----------+
|max(Volume)|
+-----------+
|  315541800|
+-----------+


scala> df.select(min ($"Volume")).show()
+-----------+
|min(Volume)|
+-----------+
|    3531300|
+-----------+
```

11)

```
scala> var a = df.filter($"Close" < 600).count()
a: Long = 1218

scala> var b = (df.filter($"High" > 500).count()*100)/1260
b: Long = 4

scala> df.select(corr("High", "Volume").alias("correlacion de Pearson")).show()
+---------------------+
|correlacion de Pearson|
+---------------------+
|  -0.20960233287942157|
+---------------------+


scala> df.groupBy(year(df("Date")).alias("Year")).max("High").sort(asc("Year")).show()
+----+------------------+
|Year|         max(High)|
+----+------------------+
|2011|120.28000300000001|
|2012|        133.429996|
|2013|        389.159988|
|2014|        489.290024|
|2015|        716.159996|
|2016|129.28999299999998|
+----+------------------+


scala> df.groupBy(month(df("Date")).alias("Month")).avg("Close").sort(asc("Month")).show()
+-----+------------------+
|Month|        avg(Close)|
+-----+------------------+
|    1|212.22613874257422|
|    2| 254.1954634020619|
|    3| 249.5825228971963|
|    4|246.97514271428562|
|    5|264.37037614150944|
|    6| 295.1597153490566|
|    7|243.64747528037387|
|    8|195.25599892727263|
|    9|206.09598121568627|
|   10|205.93297300900903|
|   11| 194.3172275445545|
|   12| 199.3700942358491|
+-----+------------------+
```

Defensa del examen

https://www.youtube.com/watch?v=l8VPDCUZynw&ab_channel=ERIKSAULRIVERAREYES