Assignment on spark

Dataframes

By Sudhanshu Arya

Task

- Let's get some quick practice with your new Spark DataFrame skills, you will be asked some basic questions about some stock market data, in this case Walmart Stock from the years 2012-2017. This exercise will just ask a bunch of questions, unlike the future machine learning exercises, which will be a little looser and be in the form of "Consulting Projects", but more on that later! For now, just answer the questions and complete the tasks below.

Load the Walmart Stock CSV File, have Spark infer the data types.

- df= spark.read.csv("dbfs:/FileStore/shared_uploads/sudhanshu,arya@emids.com/walmart_stock-2.csv",header=True,inferSchema=True) (#command)

Load the Walmart Stock CSV File, have Spark infer the data types.

In [1]:

```
1 df = spark.read.csv('walmart_stock.csv', inferSchema=True, header=True)
```

What are the column names?

- Df.show ( )
- Df.columns (#command)

```
1  df.columns
```

```
Out[7]: ['Date', 'Open', 'High', 'Low', 'Close', 'Volume', 'Adj Close']
```

Command took 0.03 seconds -- by nachanv@emids.com at 5/10/2022, 10:45:34 AM on vinisha

Cmd 11

What are the column names?

Cmd 9

```
1  df.show()
```

▸ (1) Spark Jobs

Date	Open	High	Low	Close	Volume	Adj Close
2012-01-03 00:00:00	59.97[illegible]	61.[illegible]	59.[illegible]	60.[illegible]	12[illegible]	52.[illegible]
2012-01-04 00:00:00	60.[illegible]	60.[illegible]	58.[illegible]	58.[illegible]	4[illegible]	52.[illegible]
2012-01-05 00:00:00	59.[illegible]	59.[illegible]	58.[illegible]	59.[illegible]	12[illegible]	51.[illegible]
2012-01-06 00:00:00	59.[illegible]	59.[illegible]	58.[illegible]	59.0	8[illegible]	51.[illegible]
2012-01-09 00:00:00	58.[illegible]	58.[illegible]	58.[illegible]	58.[illegible]	6[illegible]	51.[illegible]
2012-01-10 00:00:00	59.43	59.7[illegible]	58.[illegible]	59.[illegible]	8[illegible]	51.[illegible]
2012-01-11 00:00:00	59.[illegible]	59.[illegible]	59.[illegible]	59.[illegible]	6[illegible]	51.[illegible]
2012-01-12 00:00:00	59.7[illegible]	60.0	59.[illegible]	59.5	7[illegible]	51.[illegible]
2012-01-13 00:00:00	59.[illegible]	59.[illegible]	59.[illegible]	59.[illegible]	7[illegible]	51.[illegible]
2012-01-17 00:00:00	59.[illegible]	60.[illegible]	59.52	59.[illegible]	8[illegible]	52.[illegible]
2012-01-18 00:00:00	59.7[illegible]	60.[illegible]	59.[illegible]	59.[illegible]	5[illegible]	52.[illegible]
2012-01-19 00:00:00	59.93	60.73	59.75	60.[illegible]	9[illegible]	52.[illegible]
2012-01-20 00:00:00	60.75	61.25	60.[illegible]	61.[illegible]	10[illegible]	53.[illegible]
2012-01-23 00:00:00	60.[illegible]	60.[illegible]	60.[illegible]	60.91	7[illegible]	54.[illegible]
2012-01-24 00:00:00	60.75	62.0	60.75	61.[illegible]	7[illegible]	53.[illegible]
2012-01-25 00:00:00	61.10	61.[illegible]	61.[illegible]	61.[illegible]	5[illegible]	52.[illegible]
2012-01-26 00:00:00	61.[illegible]	61.84	60.77	60.[illegible]	7[illegible]	53.[illegible]
2012-01-27 00:00:00	60.[illegible]	61.[illegible]	60.[illegible]	60.[illegible]	8[illegible]	52.[illegible]

Command took 0.[illegible] seconds -- by nachanv@emids.com at 5/10/2022, 10:44:47 AM on vinisha

Cmd 10

What does the Schema look like?

Print out the first 5 columns.

Print out the first 5 columns.

Cmd 14

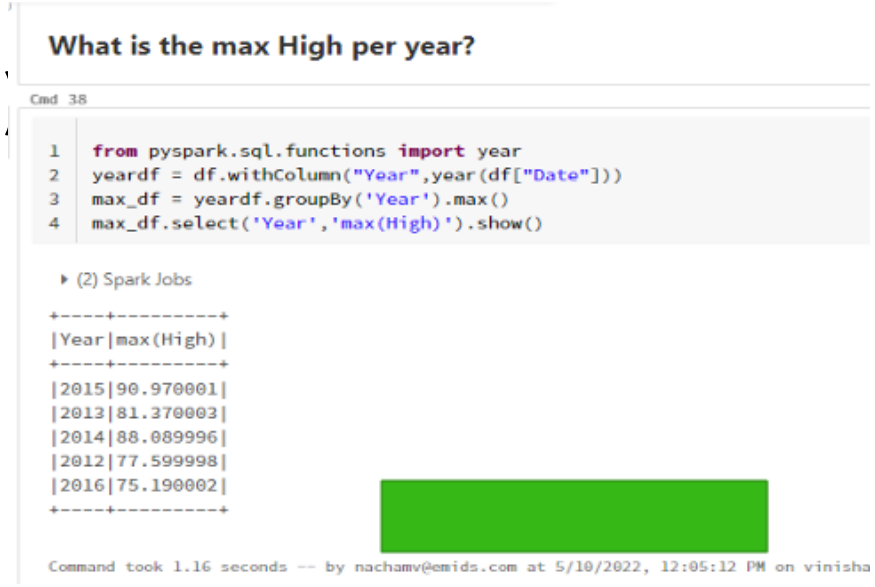
```
1  for row in df.head(5):
2      print(row)
3      print('\n')
```

▸ (1) Spark Jobs

```
Row(Date=datetime.datetime(2012, 1, 3, 0, 0), Open=59.970001, High=61.060001, Low=59.869999, Close=60.330002, Volume=12668800, Adj Close=52.619234999999996)

Row(Date=datetime.datetime(2012, 1, 4, 0, 0), Open=60.209998999999996, High=60.349998, Low=59.470001, Close=59.709998999999996, Volume=9593300, Adj Close=52.078475)

Row(Date=datetime.datetime(2012, 1, 5, 0, 0), Open=59.349998, High=59.619999, Low=58.369999, Close=59.419998, Volume=12768200, Adj Close=51.825539)

Row(Date=datetime.datetime(2012, 1, 6, 0, 0), Open=59.419998, High=59.450001, Low=58.869999, Close=59.0, Volume=8069400, Adj Close=51.45922)

Row(Date=datetime.datetime(2012, 1, 9, 0, 0), Open=59.029999, High=59.549999, Low=58.919998, Close=59.18, Volume=6679300, Adj Close=51.616215000000004)
```

What is the max High per year?

- from pyspark.sql.functions import year
- yeardf = df.withColumn("Year",year(df["Date"]))
- max_df = yeardf.groupBy('Year').max()
- max_df.select('Year','max(High)').show()



What is the average Close for each Calendar Month?

- from pyspark.sql.functions import month
- monthdf = df.withColumn("Month",month("Date"))
- monthavgs = monthdf.select("Month","Close")
- .groupBy("Month").mean()
- monthavgs.select("Month","avg(Close)").o
- rderBy('Month').show()
- (#commands)

(2) Spark Jobs

Month	avg(Close)
1	71.44801958415842
2	71.306804443299
3	71.77794377570092
4	72.97361900952382
5	72.30971688679247
6	72.4953774245283
7	74.43971943925233
8	73.02981855454546
9	72.18411785294116
10	71.57854545454543
11	72.1110893069307
12	72.84792478301885

What is the mean of the Close column?

- from pyspark.sql.functions import mean
- df.select(mean("Close")).show() (#commands)

What is the max and min of the Volume column?

- from pyspark.sql.functions import max,min
- df.select(max("Volume"),min("Volume")).show()