

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 nacham@eids.com at 5/10/2022, 10:45:34 AM on vinisha

```
What are the column names?
```

Cmd 9

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

* (7) Spark Jobs

	Date	Open	High	Low	Close	Volume	Adj Close
2012-01-02 00:00:00	50.975001	61.000001	50.000000	60.330002	12000000	52.610234000000000	
2012-01-03 00:00:00	60.200000	60.200000	59.700000	59.700000	4500000	52.878474	
2012-01-04 00:00:00	59.340000	59.610000	55.700000	59.410000	12760000	51.825520	
2012-01-05 00:00:00	59.410000	59.410001	59.000000	59.0	8000400	51.459222	
2012-01-06 00:00:00	NA.NANNA	NA.NANNA	NA.NANNA	NA.NA	NA.NANNA	NA.NANNA	
2012-01-09 00:00:00	50.43	59.70000000000000	50.00	50.040001000000004	8007300	51.404100	
2012-01-11 00:00:00	59.800001	59.520000	59.800001000000004	59.480002	8305000	51.000000	
2012-01-12 00:00:00	59.790001000000004	60.0	59.000002	59.5	7235400	51.005110000000004	
2012-01-13 00:00:00	NA.NA	NA.NANNA	NA.NANNA	NA.NANNA	NA.NANNA	NA.NANNA	
2012-01-17 00:00:00	50.000000	60.110001000000004	50.32	50.840000	8500000	52.200551	
2012-01-18 00:00:00	59.700001000000004	60.020000	59.600002	59.800007000000000	5813400	52.240121	
2012-01-19 00:00:00	59.93	60.73	59.75	60.810001000000004	9234600	52.883447	
2012-01-20 00:00:00	60.75	61.25	60.000000	61.000007000000000	10375000	53.112200000000000	
2012-01-23 00:00:00	NA.NANNA	NA.NA	NA.NANNA	NA.NA	NA.NANNA	NA.NANNA	
2012-01-24 00:00:00	60.73	61.0	60.75	61.000000000000000	7362000	53.540754000000001	
2012-01-25 00:00:00	61.10	61.000000000000004	61.000001000000004	61.470001	5815000	52.612510000000001	
2012-01-26 00:00:00	61.790000	61.80	60.77	60.970001	7435200	53.177300	
2012-01-27 00:00:00	60.500001000000004	61.100000	60.540001000000004	60.700000000000000	8207300	52.950662	

Command took 0.04 seconds -- by nacham@eids.com at 5/10/2022, 10:46:14 AM on vinisha

Cmd 10

What does the Schema look like?

What does the Schema look like?

Cmd 12

```
1 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)
```

Command took 0.02 seconds -- by nachamv@emids.com at 5/10/2022, 10:46:17 AM on vinisha

Print out the first 5 columns.

Print out the first 5 columns.

Cell 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=61.860001, Low=59.869999, Close=60.330002, Volume=12668000, 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=52.879475)

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 max High per year?

Cmd 38

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

▶ (2) Spark Jobs

```
+-----+
|Year|max(High)|
+-----+
|2015|90.970001|
|2013|81.370003|
|2014|88.089996|
|2012|77.599998|
|2016|75.190002|
+-----+
```

Command took 1.16 seconds -- by nachamv@emids.com at 5/10/2022, 12:05:12 PM on vinisha

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 mean of the Close column?
```

```
Cmd 27
```

```
1 from pyspark.sql.functions import mean
2 df.select(mean("Close")).show()
```

▶ (2) Spark Jobs

avg(Close)
72.38844998012726

Command took 0.57 seconds -- by nachamv@emids.com at 5/10/2022, 12:00:31 PM on vinisha



What is the max and min of the Volume column?

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

What is the max and min of the Volume column?

Cmd 29

```
1 from pyspark.sql.functions import max,min
```

Command took 0.04 seconds -- by nachamv@emids.com at 5/10/2022, 12:01:05 PM on vinisha

Cmd 30

```
1 df.select(max("Volume"),min("Volume")).show()
```

▶ (2) Spark Jobs

max(Volume)	min(Volume)
80898100	2094900

Command took 0.56 seconds -- by nachamv@emids.com at 5/10/2022, 12:01:12 PM on vinisha