# Spark Walmart STOCK Data Analysis Project:

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.

## Use the walmart\_stock.csv file to Answer and complete the tasks below!

**Start a simple Spark Session**

In [2]:

1

2

3

4

5

6

**import** findspark

findspark.init('/home/jubinsoni/spark-2.1.0-bin-hadoop2.7')

**from** pyspark.sql **import** SparkSession

spark = SparkSession.builder.appName('walmart').getOrCreate()

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

## What are the column names?

['Date', 'Open', 'High', 'Low', 'Close', 'Volume', 'Adj Close']

## What does the Schema look like?

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)

## Print out the first 5 columns.

## Use describe() to learn about the DataFrame.

1 df.describe().show()

+-------+------------------+-----------------+-----------------+----

-------------+-----------------+-----------------+

|  |  |  |  |
| --- | --- | --- | --- |
| |summary| | Open| | High| | Low| |
| Close| | Volume| | Adj Close| |  |

+-------+------------------+-----------------+-----------------+----

-------------+-----------------+-----------------+

| count| 1258| 1258| 1258| 1258| 1258| 1258|

| mean| 72.35785375357709|72.83938807631165| 71.9186009594594|72.3 8844998012726|8222093.481717011|67.23883848728146|

| stddev| 6.76809024470826|6.768186808159218|6.744075756255496|6.75 6859163732991| 4519780.8431556|6.722609449996857|

| min|56.389998999999996| 57.060001| 56.299999| 56.419998| 2094900| 50.363689|

| max| 90.800003| 90.970001| 89.25| 90.470001| 80898100|84.91421600000001|

+-------+------------------+-----------------+-----------------+----

-------------+-----------------+-----------------+

**Bonus Question!**

## There are too many decimal places for mean and stddev in the describe() dataframe. Format the numbers to just show up to two decimal places. Pay careful attention to the datatypes that

**.describe() returns, we didn't cover how to do this exact formatting, but we covered something very similar. Check this link for a hint (**[**http://spark.apache.org/docs/latest/api/python/pyspark.sql.html#pyspark.sql.Column.cast)**](http://spark.apache.org/docs/latest/api/python/pyspark.sql.html#pyspark.sql.Column.cast))

If you get stuck on this, don't worry, just view the solutions.

## Create a new dataframe with a column called HV Ratio that is the ratio of the High Price versus volume of stock traded for a day.

## What day had the Peak High in Price?

+-----------------+

## What is the max and min of the Volume column?

## How many days was the Close lower than 60 dollars?

## What percentage of the time was the High greater than 80 dollars ? In other words, (Number of Days High>80)/(Total Days in the dataset)

## What is the max High per year?

## In other words, across all the years, what is the average Close price for Jan,Feb, Mar, etc... Your result will have a value for each of these months.

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.

## Use the walmart\_stock.csv file to Answer and complete the tasks below!

**Start a simple Spark Session**

In [2]:

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

In [1]:

## What are the column names?

In [2]:

Out[2]:

['Date', 'Open', 'High', 'Low', 'Close', 'Volume', 'Adj Close']

## What does the Schema look like?

|  |  |  |
| --- | --- | --- |
| 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)

## Print out the first 5 columns.

In [4]:

Row(Date=datetime.datetime(2012, 1, 3, 0, 0), Open=59.970001, High=6 1.060001, Low=59.869999, Close=60.330002, Volume=12668800, Adj Close

=52.619234999999996)

Row(Date=datetime.datetime(2012, 1, 4, 0, 0), Open=60.20999899999999 6, High=60.349998, Low=59.470001, Close=59.709998999999996, Volume=9

593300, Adj Close=52.078475)

Row(Date=datetime.datetime(2012, 1, 5, 0, 0), Open=59.349998, High=5 9.619999, 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=5 9.450001, Low=58.869999, Close=59.0, Volume=8069400, Adj Close=51.45

922)

Row(Date=datetime.datetime(2012, 1, 9, 0, 0), Open=59.029999, High=5 9.549999, Low=58.919998, Close=59.18, Volume=6679300, Adj Close=51.6

16215000000004)

## Use describe() to learn about the DataFrame.

+-------+------------------+-----------------+-----------------+----

-------------+-----------------+-----------------+

|  |  |  |  |
| --- | --- | --- | --- |
| |summary| | Open| | High| | Low| |
| Close| | Volume| | Adj Close| |  |

+-------+------------------+-----------------+-----------------+----

-------------+-----------------+-----------------+

| count| 1258| 1258| 1258| 1258| 1258| 1258|

| mean| 72.35785375357709|72.83938807631165| 71.9186009594594|72.3 8844998012726|8222093.481717011|67.23883848728146|

| stddev| 6.76809024470826|6.768186808159218|6.744075756255496|6.75 6859163732991| 4519780.8431556|6.722609449996857|

| min|56.389998999999996| 57.060001| 56.299999| 56.419998| 2094900| 50.363689|

| max| 90.800003| 90.970001| 89.25| 90.470001| 80898100|84.91421600000001|

+-------+------------------+-----------------+-----------------+----

-------------+-----------------+-----------------+

**Bonus Question!**

## There are too many decimal places for mean and stddev in the describe() dataframe. Format the numbers to just show up to two decimal places. Pay careful attention to the datatypes that

**.describe() returns, we didn't cover how to do this exact formatting, but we covered something very similar. Check this link for a hint (**[**http://spark.apache.org/docs/latest/api/python/pyspark.sql.html#pyspark.sql.Column.cast)**](http://spark.apache.org/docs/latest/api/python/pyspark.sql.html#pyspark.sql.Column.cast))

If you get stuck on this, don't worry, just view the solutions.

In [25]:

|  |  |  |
| --- | --- | --- |
| 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)

+-------+--------+--------+--------+--------+----------+

|summary| Open| High| Low| Close| Volume|

+-------+--------+--------+--------+--------+----------+

| count|1,258.00|1,258.00|1,258.00|1,258.00| 1,258|

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| | | mean| | 72.36| | 72.84| | 71.92| | 72.39| | 8,222,093| |
| | | stddev| | 6.77| | 6.77| | 6.74| | 6.76| | 4,519,781| |
| | | min| | 56.39| | 57.06| | 56.30| | 56.42| | 2,094,900| |

| max| 90.80| 90.97| 89.25| 90.47|80,898,100|

+-------+--------+--------+--------+--------+----------+

## Create a new dataframe with a column called HV Ratio that is the ratio of the High Price versus volume of stock traded for a day.

In [46]:

+--------------------+

| HV Ratio|

+--------------------+

|4.819714653321546E-6|

|6.290848613094555E-6|

|4.669412994783916E-6|

|7.367338463826307E-6|

|8.915604778943901E-6|

|8.644477436914568E-6|

|9.351828421515645E-6|

| 8.29141562102703E-6|

|7.712212102001476E-6|

|7.071764823529412E-6|

|1.015495466386981E-5|

|6.576354146362592...|

| 5.90145296180676E-6|

|8.547679455011844E-6|

|8.420709512685392E-6|

|1.041448341728929...|

## What day had the Peak High in Price?

Out[61]:

datetime.datetime(2015, 1, 13, 0, 0)

## What is the mean of the Close column?

In [90]:

+-----------------+

| avg(Close)|

+-----------------+

|72.38844998012726|

+-----------------+

## What is the max and min of the Volume column?

In [92]:

+-----------+-----------+

|max(Volume)|min(Volume)|

+-----------+-----------+

| 80898100| 2094900|

+-----------+-----------+

## How many days was the Close lower than 60 dollars?

In [101]:

Out[101]:

81

## What percentage of the time was the High greater than 80 dollars ? In other words, (Number of Days High>80)/(Total Days in the dataset)

In [105]:

Out[105]:

9.141494435612083

## What is the Pearson correlation between High and Volume?

**Hint (**[**http://spark.apache.org/docs/latest/api/python/pyspark.sql.html#pyspark.sql.DataFrameStatFunction**](http://spark.apache.org/docs/latest/api/python/pyspark.sql.html#pyspark.sql.DataFrameStatFunction)

In [113]:

Out[113]:

-0.3384326061737161

In [114]:

+-------------------+

| corr(High, Volume)|

+-------------------+

|-0.3384326061737161|

+-------------------+

## What is the max High per year?

In [133]:

+----+---------+

|Year|max(High)|

+----+---------+

|2015|90.970001|

|2013|81.370003|

|2014|88.089996|

|2012|77.599998|

|2016|75.190002|

+----+---------+

## What is the average Close for each Calendar Month?

**In other words, across all the years, what is the average Close price for Jan,Feb, Mar, etc... Your result will have a value for each of these months.**

In [139]:

+-----+-----------------+

|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| |

+-----+-----------------+

# Thank you!