StudentID:311511056

Name:游翔竣

Checkpoint1

```
for row in df.head(5):
    print(row)
```

0.2s Python

```
Row(Date='2012-01-03', Open=59.970001, High=61.060001, Low=59.869999, Close=60.330002, Volume=12668800, Adj Close=52.619234999999996)
Row(Date='2012-01-04', Open=60.209989999999996, High=60.349998, Low=59.470001, Close=59.709989999999996, Volume=9593300, Adj Close=52.078475)
Row(Date='2012-01-05', Open=59.349998, High=59.619999, Low=58.369999, Close=59.419998, Volume=12768200, Adj Close=51.825539)
Row(Date='2012-01-06', Open=59.419998, High=59.450001, Low=58.869999, Close=59.0, Volume=8069400, Adj Close=51.45922)
Row(Date='2012-01-09', Open=59.029999, High=59.549999, Low=58.919998, Close=59.18, Volume=6679300, Adj Close=51.616215000000004)
```

Checkpoint2

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

0.3s Python

```
+-------+----------+------------------+------------------+------------------+------------------+-----------------+------------------+
|summary|      Date|              Open|              High|               Low|             Close|           Volume|         Adj Close|
+-------+----------+------------------+------------------+------------------+------------------+-----------------+------------------+
|  count|      1258|              1258|              1258|              1258|              1258|             1258|              1258|
|   mean|      null| 72.35785375357709|72.83938807631165| 71.9186009594594|72.38844998012726|8222093.481717011|67.23883848728146|
| stddev|      null|  6.768090244708226|6.768186808159218|6.744075756255496|6.756859163732991|  4519780.8431556|6.722609449996857|
|    min|2012-01-03|56.389998999999996|        57.060001|        56.299999|        56.419998|          2094900|        50.363689|
|    max|2016-12-30|         90.800003|         90.970001|           89.25|         90.470001|         80898100|84.91421600000001|
+-------+----------+------------------+------------------+------------------+------------------+-----------------+------------------+
```

Checkpoint3

```
result = df.describe()
result.select(result['summary'],
              format_number(result['Open'].cast('float'),2).alias('Open'),
              format_number(result['High'].cast('float'),2).alias('High'),
              format_number(result['Low'].cast('float'),2).alias('Low'),
              format_number(result['Close'].cast('float'),2).alias('Close'),
              result['Volume'].cast(IntegerType()).alias('Volume'),
             ).show()
```

0.3s Python

```
+-------+--------+--------+--------+--------+--------+
|summary|    Open|    High|     Low|   Close|  Volume|
+-------+--------+--------+--------+--------+--------+
|  count|1,258.00|1,258.00|1,258.00|1,258.00|    1258|
|   mean|   72.36|   72.84|   71.92|   72.39| 8222093|
| stddev|    6.77|    6.77|    6.74|    6.76| 4519780|
|    min|   56.39|   57.06|   56.30|   56.42| 2094900|
|    max|   90.80|   90.97|   89.25|   90.47|80898100|
+-------+--------+--------+--------+--------+--------+
```

Checkpoint4

```
newdf = df.withColumn("HV Ratio", df["High"]/df["Volume"])
newdf.select(newdf['HV Ratio'].alias('HV Ratio')).show()
```

0.2s Python

```
+--------------------+
|            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|
|1.0154954663869 81E-5|
|6.576354146362592...|
| 5.90145296180676E-6|
|8.547679455011844E-6|
|8.428709512685392E-6|
|1.041448341728929...|
|8.316075414862431E-6|
|9.721183814992126E-6|
|8.029436027707578E-6|
|6.307432259386365E-6|
+--------------------+
only showing top 20 rows
```

Checkpoint5

```
# Could have also used describe
from pyspark.sql.functions import mean,max,min
```

✓ 0.7s Python

```
df.select(
        mean("Close"),
        max("Close"),
        min("Close")
).show()
```

✓ 0.2s Python

```
+-----------------+----------+----------+
|      avg(Close)|max(Close)|min(Close)|
+-----------------+----------+----------+
|72.38844998012726| 90.470001| 56.419998|
+-----------------+----------+----------+
```

Checkpoint6

```
from pyspark.sql.functions import count # hint
result = df.filter(df['Close'] < 60)
result.select(count("Close")).show()
```

✓ 0.2s Python

```
+------------+
|count(Close)|
+------------+
|          81|
+------------+
```

Checkpoiont7

```
# Many ways to do this
result = df.filter(df['High'] > 80)
result.count() / df.count() * 100
```

✓ 0.3s Python

```
9.141494435612083
```

Checkpoint8

```
from pyspark.sql.functions import corr # hint
df.select(corr("High", "Volume")).show()
```

✓ 0.2s Python

```
+-------------------+
| corr(High, Volume)|
+-------------------+
|-0.3384326061737161|
+-------------------+
```

Checkpoint9

```
from pyspark.sql.functions import month
# hint
monthdf = df.withColumn("Month",month("Date"))
avg_df = monthdf.groupBy('Month').mean()
avg_df.sort('Month').select('Month','avg(Close)').show()
```

✓ 0.5s Python

```
+-----+-----------------+
|Month|       avg(Close)|
+-----+-----------------+
|    1|71.44801958415842|
|    2| 71.30680444329 9|
|    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|
+-----+-----------------+
```