SPARK JOB FINISHED

```
import org.apache.hadoop.fs.{FileSystem, Path}
import org.apache.spark.sql.functions._
import org.apache.spark.sql.{Row, SparkSession}

val directoryPath = "/user/sb9509_nyu_edu/stocks"
val fs = FileSystem.get(spark.sparkContext.hadoopConfiguration)
val stockFiles = fs.listStatus(new Path(directoryPath)).filter(_.getPath.getName.endsWith(".csv")).map(_.getPath.toString)

val results = stockFiles.map { filePath =>
  val stockName = filePath.split("/").last.stripSuffix(".L.csv")

  val rawDf = spark.read.option("header", false).option("inferSchema", "true").csv(filePath)
  val filteredRDD = rawDf.rdd.zipWithIndex().filter { case (_, idx) => idx >= 3 }.map(_._1)
  val filteredDf = spark.createDataFrame(filteredRDD, rawDf.schema)
  val columnNames = Seq("Date", "AdjClose", "Close", "Open", "High", "Low", "Volume")
  val finalDf = filteredDf.toDF(columnNames: _*)
  val selectedDf = finalDf.select($"Date", $"Close").withColumn("Date", to_date($"Date", "yyyy-MM-dd")).withColumn("Close", $"Close".cast("double

  val startDate = selectedDf.agg(min("Date")).collect().head.getDate(0)
  val minValue = selectedDf.agg(min("Close")).collect().head.getDouble(0)
  val maxValue = selectedDf.agg(max("Close")).collect().head.getDouble(0)
  val nullCount = selectedDf.filter($"Close".isNull).count()
  val stdDevRow = selectedDf.agg(stddev("Close")).collect().head
  val stdDev = if (stdDevRow.isNullAt(0)) Double.NaN else stdDevRow.getDouble(0)


  (stockName, startDate, minValue, maxValue, nullCount, stdDev)
}
```

Took 1 min 38 sec. Last updated by sb9509_nyu_edu at November 22 2024, 5:59:55 PM.

FINISHED

```
//Profiling

import spark.implicits._
val resultsDf = results.toSeq.toDF("StockSymbol", "StartDate", "MinValue", "MaxValue", "NullCount", "StdDev")

val valueDistribution = resultsDf.select("*")
valueDistribution.createOrReplaceTempView("value_distribution")
```

Took 1 sec. Last updated by sb9509_nyu_edu at November 22 2024, 6:02:52 PM. (outdated)

FINISHED

```
%sql
SELECT StockSymbol, StartDate, MaxValue, MinValue, NullCount, StdDev FROM value_distribution
```

settings ▾

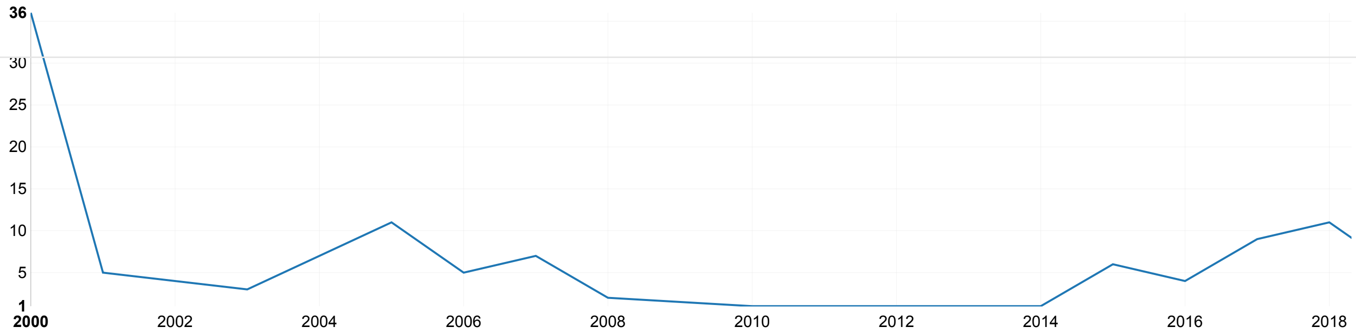
StockSymbol ▲	StartDate	MaxValue	MinValue	NullCount
3IN	2007-03-08	366.5	109.0739974975586	0
450	2017-08-17	6.560299873352051	0.017999999225139618	0
4BB	2021-02-17	1820.0	310.0	0
4GBL	2021-12-07	95.0	43.5	0
70GD	2000-01-04	70.5	0.5	0
78GL	2000-01-04	1.159999966621399	0.6000000238418579	0
79GL	2000-01-04	1.559999942779541	0.75	0
80M	2005-03-31	500.0	0.26499998569488525	0

Took 0 sec. Last updated by sb9509_nyu_edu at November 22 2024, 6:02:53 PM.

SPARK JOB FINISHED

```
%sql
SELECT
    YEAR(CAST(StartDate AS DATE)) AS Year,
    COUNT(DISTINCT StockSymbol) AS StockCount
FROM value_distribution
WHERE YEAR(CAST(StartDate AS DATE)) BETWEEN 2000 AND 2024
GROUP BY Year
ORDER BY Year
```

settings ▾



Took 0 sec. Last updated by sb9509_nyu_edu at November 22 2024, 6:02:59 PM. (outdated)

SPARK JOB FINISHED

```
import org.apache.spark.sql.functions._
import org.apache.spark.sql.expressions.Window


stockFiles.foreach { filePath =>
  val stockName = filePath.split("/").last.stripSuffix(".L.csv")

  val rawDf = spark.read.option("header", false).option("inferSchema", "true").csv(filePath)
  val filteredRDD = rawDf.rdd.zipWithIndex().filter { case (_, idx) => idx >= 3 }.map(_._1)
  val filteredDf = spark.createDataFrame(filteredRDD, rawDf.schema)
  val columnNames = Seq("Date", "AdjClose", "Close", "Open", "High", "Low", "Volume")
  val finalDf = filteredDf.toDF(columnNames: _*)

  val selectedDf = finalDf.select($"Date", $"Close").withColumn("Date", to_date($"Date", "yyyy-MM-dd")).withColumn("Close", $"Close".cast("double


  val forwardFillSpec = Window.orderBy("Date").rowsBetween(Window.unboundedPreceding, 0)
  val backwardFillSpec = Window.orderBy("Date").rowsBetween(0, Window.unboundedFollowing)

  val cleanedDf = selectedDf.withColumn("Close", last($"Close", ignoreNulls = true).over(forwardFillSpec)).withColumn("Close", coalesce($"Close",
      (backwardFillSpec)))

  cleanedDf.write.option("header", "true").mode("overwrite").csv(s"/user/sb9509_nyu_edu/stocks_cleaned/$stockName")
}
```

Took 2 min 30 sec. Last updated by sb9509_nyu_edu at November 22 2024, 6:30:18 PM.

READY