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

val directoryPath = "/user/ag9563_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(".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("d

  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)
}
```

SPARK JOB RUNNING 0%

Started an hour ago.

```
//Profiling

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

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

FINISHED

```
import spark.implicits._
resultsDf: org.apache.spark.sql.DataFrame = [StockSymbol: string, StartDate: date ... 4 more fields]
valueDistribution: org.apache.spark.sql.DataFrame = [StockSymbol: string, StartDate: date ... 4 more fields]
```

Took 1 sec. Last updated by ag9563_nyu_edu at November 22 2024, 8:25:14 PM.

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

FINISHED

settings

StockSymbol	StartDate	MaxValue	MinValue	NullCount
ACAD	2004-05-27	57.0	0.6600000262260437	0
ACET	2018-01-26	145.9499969482422	1.0700000524520874	0
ACFC	2004-10-05	101.98979187011719	0.8700000047683716	0
ACFN	2000-01-03	13.600000381469727	0.07999999821186066	0
ACGL	2000-01-03	114.86000061035156	1.2708330154418945	0
ACHC	2000-01-03	89.05999755859375	0.5	0
ACIW	2000-01-03	52.869998931884766	1.6666669845581055	0
ACLS	2000-07-11	200.47999572753906	0.6800000071525574	0

Took 0 sec. Last updated by ag9563_nyu_edu at November 22 2024, 8:25:23 PM.

```
%sql
SELECT
    StockSymbol,
    StartDate,
    MaxValue,
    MinValue,
    NullCount,
    StdDev
FROM value_distribution
ORDER BY MaxValue DESC
LIMIT 10;
```

SPARK JOB (http://nyu-dataproc-sw-8w9r.c.hpc-dataproc-19b8.internal:34641/jobs/job?id=9239) FINISHED

settings

StockSymbol	StartDate	MaxValue	MinValue	NullCo
ASTI	2022-08-24	181900.0	2.255000114440918	0
BPTH	2008-03-04	22960.0	0.8500000238418579	0
HCT	2000-01-03	20200.0	0.0010000000474974513	0
CASI	2000-01-03	10835.0	1.4800000190734863	0
ASTC	2000-01-03	9609.375	7.119999885559082	0
BLIN	2007-06-29	6312.5	0.6299999952316284	0
AGEN	2000-02-08	6197.25244140625	4.25	0
ACTA	2000-01-03	4001.25	0.7113999724388123	0

Took 1 sec. Last updated by ag9563_nyu_edu at November 22 2024, 8:28:36 PM. (outdated)

```
%sql
SELECT
    StockSymbol,
    StartDate,
    MaxValue,
    MinValue,
    NullCount,
    StdDev
FROM value_distribution
ORDER BY MinValue ASC
LIMIT 10;
```

SPARK JOB (http://nyu-dataproc-sw-8w9r.c.hpc-dataproc-19b8.internal:34641/jobs/job?id=9240) FINISHED

settings

StockSymbol	StartDate	MaxValue	MinValue	NullCo
ANTH	2010-03-01	553.5999755859375	9.999999747378752E-6	0
BSPM	2008-12-18	113.19022369384766	9.999999747378752E-6	0
CACH	2000-01-03	21.92232894897461	4.999999873689376E-5	0
ACUR	2000-01-03	164.0	9.999999747378752E-5	0
BGMD	2011-02-04	41.599998474121094	9.999999747378752E-5	0
AMCF	2010-01-26	7.909999847412109	9.999999747378752E-5	0
CALA	2014-10-02	597.0	9.999999747378752E-5	0
AURX	2017-10-24	4.420000076293945	9.999999747378752E-5	0

Took 1 sec. Last updated by ag9563_nyu_edu at November 22 2024, 8:29:13 PM. (outdated)

```
%sql
SELECT
    StockSymbol,
    StartDate,
    MaxValue,
    MinValue,
    NullCount,
    StdDev
FROM value_distribution
ORDER BY StdDev DESC
LIMIT 10;
```

SPARK JOB (http://nyu-dataproc-sw-8w9r.c.hpc-dataproc-19b8.internal:34641/jobs/job?id=9241) FINISHED

settings

StockSymbol	StartDate	MaxValue	MinValue	NullCo
CLRB	2005-11-10	1.0251E7	1.2699999809265137	0
BIOL	2000-01-03	229525.6875	0.006099999882280827	0
ASTI	2022-08-24	181900.0	2.255000114440918	0
HCT	2000-01-03	20200.0	0.0010000000474974513	0
BPTH	2008-03-04	22960.0	0.8500000238418579	0
ASTC	2000-01-03	9609.375	7.119999885559082	0
BLIN	2007-06-29	6312.5	0.6299999952316284	0
CASI	2000-01-03	10835.0	1.4800000190734863	0

Took 0 sec. Last updated by ag9563_nyu_edu at November 22 2024, 8:29:47 PM.

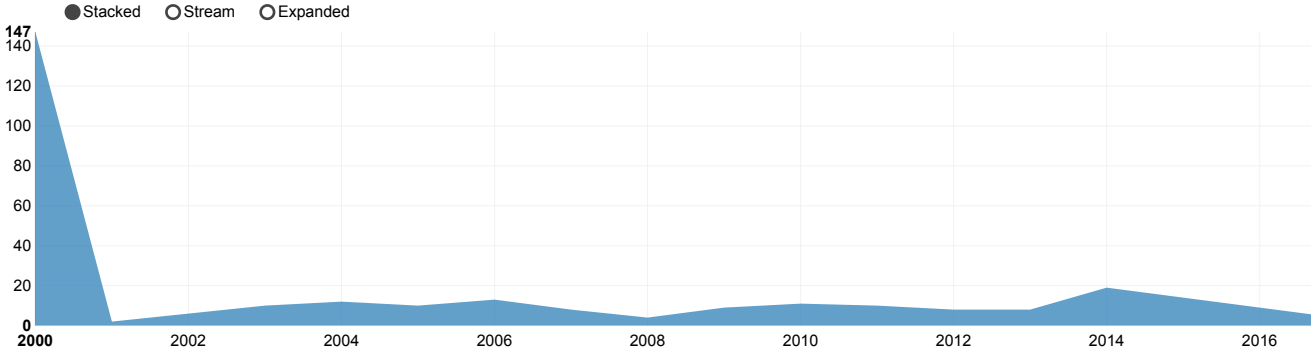
```
%sql
SELECT
```

SPARK JOB FINISHED

```
    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

Stacked Stream Expanded



Took 0 sec. Last updated by ag9563_nyu_edu at November 22 2024, 8:25:46 PM.

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(".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("d


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

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

  cleanedDf.write.option("header", "true").mode("overwrite").csv(s"/user/ag9563_nyu_edu/stocks_cleaned/$stockName")
}
```

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

Took 13 sec. Last updated by ag9563_nyu_edu at November 22 2024, 7:39:18 PM.

READY