1

```
//Analysis of Harddrive data from Backblaze
```

READY

```
import org.apache.spark.sql.SparkSession
import org.apache.spark.sql.functions._
import org.apache.spark.sql.expressions.Window
val spark=SparkSession.builder().appName("Spark dataframe eaxample").config("spark.some.config.option"
import spark.implicits._


var data=spark.read.option("sep",",").option("header","true").csv("C:/BigData/~data/FP-Data-2020/Drive
```

FINISHED

```
import org.apache.spark.sql.SparkSession
import org.apache.spark.sql.functions._
import org.apache.spark.sql.expressions.Window
spark: org.apache.spark.sql.SparkSession = org.apache.spark.sql.SparkSession@2f8cae9b
import spark.implicits._
data: org.apache.spark.sql.DataFrame = [date: string, serial_number: string ... 147 more fields]
```

Took 33 sec. Last updated by admin at May 04 2021, 6:53:41 AM.

```
//count of lines
val linecount= data.count()
```

FINISHED

```
linecount: Long = 52286398
```

Took 1 min 17 sec. Last updated by admin at May 04 2021, 6:56:32 AM.

```
//Parquet implementation

data.write.parquet("C:/BigData/~data/project.parquet")
```

ERROR

Took 1 sec. Last updated by admin at May 04 2021, 6:56:38 AM.

```
//Reading values from parquet

val parquetProject = spark.read.parquet("C:/BigData/~data/project.parquet")
```

FINISHED

```
parquetProject: org.apache.spark.sql.DataFrame = [date: string, serial_number: string ... 147 more fie
lds]
```

Took 1 sec. Last updated by admin at May 04 2021, 6:56:48 AM.

```
//selecting most used columns in the data
val data_useful = parquetProject.select(
    col("date"),
    col("serial_number"),
    col("model"),
    col("capacity_bytes"),
    col("failure"),
    col("smart_1_raw"),col("smart_5_raw"),col("smart_9_raw"),col("smart_187_raw"),
    col("smart_188_raw"),col("smart_194_raw"),col("smart_197_raw"),
```

FINISHED

```
    col("smart_198_raw"))
```

```
+----------+-------------+-------------+--------------+-------+-----------+-----------+-----------+-
------------+-------------+-------------+-------------+-------------+
|      date|serial_number|        model|capacity_bytes|failure|smart_1_raw|smart_5_raw|smart_9_raw|s
mart_187_raw|smart_188_raw|smart_194_raw|smart_197_raw|smart_198_raw|
+----------+-------------+-------------+--------------+-------+-----------+-----------+-----------+-
------------+-------------+-------------+-------------+-------------+
|2020-12-01|     ZLW0EGC6|ST12000NM001G|12000138625024|      0|161570952.0|        0.0|     4322.0|
0.0|          0.0|         29.0|          0.0|          0.0|
|2020-12-01|     Z305B2QN|  ST4000DM000| 4000787030016|      0|144476168.0|        0.0|    43504.0|
0.0|          0.0|         22.0|          0.0|          0.0|
|2020-12-01|     ZJV0XJQ4|ST12000NM0007|12000138625024|      0|  1030832.0|        0.0|    20528.0|
0.0|          0.0|         28.0|          0.0|          0.0|
|2020-12-01|     ZJV0XJQ3|ST12000NM0007|12000138625024|      0|177532056.0|        0.0|    17582.0|
0.0|          0.0|         33.0|          0.0|          0.0|
|2020-12-01|     ZLW18MKT|ST14000NM001G|14000519643136|      0|227599440.0|        0.0|      422.0|
0.0|          0.0|         25.0|          0.0|          0.0|
+----------+-------------+-------------+--------------+-------+-----------+-----------+-----------+-
------------+-------------+-------------+-------------+-------------+
```

Took 3 sec. Last updated by admin at May 04 2021, 6:56:56 AM.

FINISHED

```
//Adding an additional column for manufacturer based on model
import org.apache.spark.sql.functions.{when, _}

val data_MFG = data_useful.withColumn("manufacturer", when(col("model") like "ST%","Seagate")
      .when(col("model") like "Hitachi%","HGST/Hitachi")
      .when(col("model") like "HGST%","HGST/Hitachi")
      .when(col("model") like "TOSHI%","TOSHIBHA")
      .when(col("model") like "Seagate%","Seagate")
      .when(col("model") like "Dell%","Dell")
      .when(col("model") like "WDC%","WDC")
      .otherwise("Unknown"))

data_MFG.show(5)
```

```
+----------+-------------+-------------+--------------+-------+-----------+-----------+-----------+-
------------+-------------+-------------+-------------+-------------+------------+
|      date|serial_number|        model|capacity_bytes|failure|smart_1_raw|smart_5_raw|smart_9_raw|s
mart_187_raw|smart_188_raw|smart_194_raw|smart_197_raw|smart_198_raw|manufacturer|
+----------+-------------+-------------+--------------+-------+-----------+-----------+-----------+-
------------+-------------+-------------+-------------+-------------+------------+
|2020-12-01|     ZLW0EGC6|ST12000NM001G|12000138625024|      0|161570952.0|        0.0|     4322.0|
0.0|          0.0|         29.0|          0.0|          0.0|     Seagate|
|2020-12-01|     Z305B2QN|  ST4000DM000| 4000787030016|      0|144476168.0|        0.0|    43504.0|
0.0|          0.0|         22.0|          0.0|          0.0|     Seagate|
|2020-12-01|     ZJV0XJQ4|ST12000NM0007|12000138625024|      0|  1030832.0|        0.0|    20528.0|
0.0|          0.0|         28.0|          0.0|          0.0|     Seagate|
|2020-12-01|     ZJV0XJQ3|ST12000NM0007|12000138625024|      0|177532056.0|        0.0|    17582.0|
0.0|          0.0|         33.0|          0.0|          0.0|     Seagate|
|2020-12-01|     ZLW18MKT|ST14000NM001G|14000519643136|      0|227599440.0|        0.0|      422.0|
0.0|          0.0|         25.0|          0.0|          0.0|     Seagate|
+----------+-------------+-------------+--------------+-------+-----------+-----------+-----------+-
------------+-------------+-------------+-------------+-------------+------------+
```

Took 1 sec. Last updated by admin at May 04 2021, 6:57:03 AM.

FINISHED

```
//To show Capacity bytes in more user friendly
```

```
val data_mod = data_MFG.withColumn("capacity_bytes_TB", when(col("capacity_bytes") like "-1%","not_det
      .when(col("capacity_bytes") like "12000138625024%","12TB")
      .when(col("capacity_bytes") like "500107862016%","500GB")
      .when(col("capacity_bytes") like "480036847616%","480GB")
      .when(col("capacity_bytes") like "480103981056%","480GB")
      .when(col("capacity_bytes") like "240057409536%","240GB")
      .when(col("capacity_bytes") like "250059350016%","240GB")
      .when(col("capacity_bytes") like "14000519643136%","14TB")
      .when(col("capacity_bytes") like "6001175126016%","6TB")
      .when(col("capacity_bytes") like "4000787030016%","4TB")
      .when(col("capacity_bytes") like "16000900661248%","16TB")
      .when(col("capacity_bytes") like "10000831348736%","10TB")
      .when(col("capacity_bytes") like "8001563222016%","8TB")
      .when(col("capacity_bytes") like "2000398934016%","2TB")
      .when(col("capacity_bytes") like "18000207937536","18TB")
      .when(col("capacity_bytes") like "1000204886016%","1TB"))

data_mod.show(5)
```

1

```
mart_187_raw|smart_188_raw|smart_194_raw|smart_197_raw|smart_198_raw|manufacturer|capacity_bytes_TB|
+----------+-------------+-------------+-------------+-------+-----------+-----------+-----------+-
------------+-------------+-------------+-------------+-------------+------------+-----------------+
|2020-12-01|     ZLW0EGC6|ST12000NM001G|12000138625024|      0|161570952.0|        0.0|     4322.0|
0.0|          0.0|         29.0|          0.0|          0.0|     Seagate|            12TB|
|2020-12-01|     Z305B2QN|  ST4000DM000| 4000787030016|      0|144476168.0|        0.0|    43504.0|
0.0|          0.0|         22.0|          0.0|          0.0|     Seagate|             4TB|
|2020-12-01|     ZJV0XJQ4|ST12000NM0007|12000138625024|      0|  1030832.0|        0.0|    20528.0|
0.0|          0.0|         28.0|          0.0|          0.0|     Seagate|            12TB|
|2020-12-01|     ZJV0XJQ3|ST12000NM0007|12000138625024|      0|177532056.0|        0.0|    17582.0|
0.0|          0.0|         33.0|          0.0|          0.0|     Seagate|            12TB|
|2020-12-01|     ZLW18MKT|ST14000NM001G|14000519643136|      0|227599440.0|        0.0|      422.0|
0.0|          0.0|         25.0|          0.0|          0.0|     Seagate|            14TB|
+----------+-------------+-------------+-------------+-------+-----------+-----------+-----------+-
------------+-------------+-------------+-------------+-------------+------------+-----------------+
only showing top 5 rows

data_mod: org.apache.spark.sql.DataFrame = [date: string, serial_number: string ... 13 more fields]
```

Took 0 sec. Last updated by admin at May 04 2021, 6:57:08 AM.

FINISHED

```
//Handling null values to avoid inconsistency
//val df = data_mod.na.drop(Seq("smart_194_raw"))
val df= data_mod.na.fill(0)
```

```
df: org.apache.spark.sql.DataFrame = [date: string, serial_number: string ... 13 more fields]
```

Took 1 sec. Last updated by admin at May 04 2021, 6:57:19 AM.

FINISHED

```
//creating a temporary table for the dataframe
df.createOrReplaceTempView("query_data")
```

Took 1 sec. Last updated by admin at May 04 2021, 6:57:22 AM.

FINISHED

```
//Drive temperature
val drive_temp = spark.sql("""
select
    manufacturer,
    model,
    capacity_bytes_TB,
```

```
    max(failure) as failure,
    Avg(smart_194_raw) as temperature
from query_data
group by manufacturer,model, capacity_bytes_TB
order by manufacturer, capacity_bytes_TB
""").cache()
```

1

```
drive_temp: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [manufacturer: string, model: str
ing ... 3 more fields]
```

Took 1 sec. Last updated by admin at May 04 2021, 6:57:26 AM.

```
drive_temp.show(5)
```

FINISHED

```
+------------+--------------------+-----------------+-------+------------------+
|manufacturer|               model|capacity_bytes_TB|failure|       temperature|
+------------+--------------------+-----------------+-------+------------------+
|HGST/Hitachi|HGST HUH721010ALE600|             10TB|      0| 5.041808990299221|
|HGST/Hitachi|HGST HUH721212ALE600|             12TB|      1| 9.833429642850177|
|HGST/Hitachi|HGST HUH721212ALE604|             12TB|      1|23.008064398842333|
|HGST/Hitachi|HGST HUH721212ALN604|             12TB|      1| 8.398613210735169|
|HGST/Hitachi|HGST HMS5C4040ALE640|              4TB|      1| 8.418222455591843|
+------------+--------------------+-----------------+-------+------------------+
only showing top 5 rows
```

Took 38 sec. Last updated by admin at May 04 2021, 6:59:16 AM.

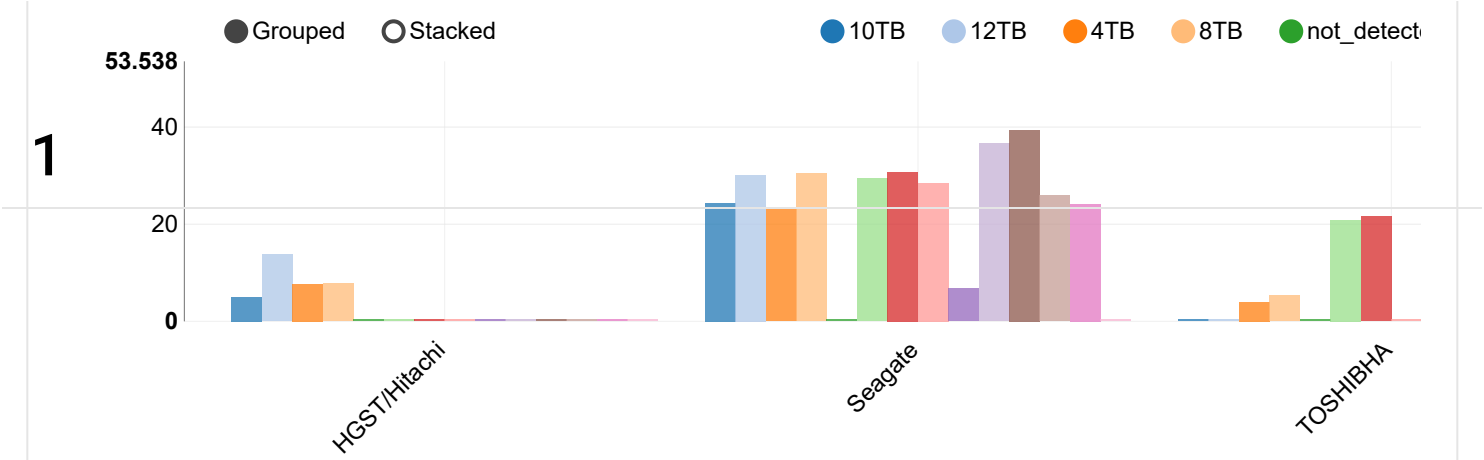
```
drive_temp.createOrReplaceTempView("temperature")
```

FINISHED

Took 0 sec. Last updated by admin at May 04 2021, 6:59:23 AM.

```
%sql
select * from temperature
```

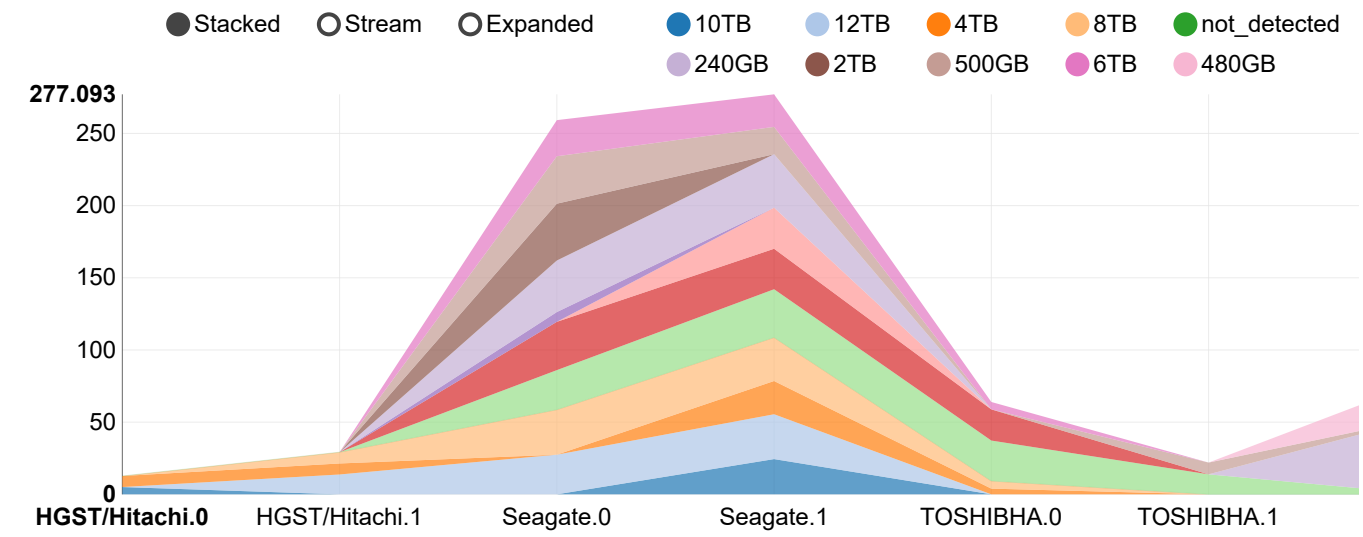
FINISHED



Took 0 sec. Last updated by admin at May 04 2021, 7:06:55 AM. (outdated)

```
%sql
select * from temperature
```

FINISHED



Took 0 sec. Last updated by admin at May 04 2021, 6:59:26 AM. (outdated)

```
//Calculating Annual Failue rate
```

FINISHED

Took 0 sec. Last updated by admin at May 01 2021, 3:53:17 PM. (outdated)

```
//drivedays = the number of days all of the drives being observed were operational during the period
val drivedays = spark.sql("SELECT manufacturer, model, capacity_bytes_TB, count(date) AS drive_days "+
                          "FROM query_data "+
```

FINISHED

```
                                "GROUP BY model, manufacturer, capacity_bytes_TB")
drivedays.show(5)
```

```
+------------+--------------------+-----------------+----------+
|manufacturer|               model|capacity_bytes_TB|drive_days|
+------------+--------------------+-----------------+----------+
|HGST/Hitachi|HGST HUH721212ALE600|             12TB|    820272|
|     Seagate|       ST12000NM0008|     not_detected|      3963|
|     Seagate|         ST4000DM000|              4TB|   6982962|
|HGST/Hitachi|HGST HUH721212ALE604|     not_detected|       742|
|         WDC| WDC WUH721414ALE6L4|             14TB|    226848|
+------------+--------------------+-----------------+----------+
only showing top 5 rows

drivedays: org.apache.spark.sql.DataFrame = [manufacturer: string, model: string ... 2 more fields]
```

Took 9 sec. Last updated by admin at May 04 2021, 7:01:00 AM.

```
//Drive Failures is the number of drives that failed during the period of observation
val failure_bymodel = spark.sql("SELECT manufacturer, model,capacity_bytes_TB, count(*) AS failures, a
                                "FROM query_data "+
                                "WHERE failure = 1 "+
                                "GROUP BY model, manufacturer, capacity_bytes_TB")
failure_bymodel.show(5)
```

FINISHED

```
+------------+--------------------+-----------------+--------+------------------+
|manufacturer|               model|capacity_bytes_TB|failures|       temperature|
+------------+--------------------+-----------------+--------+------------------+
|HGST/Hitachi|HGST HUH721212ALE600|             12TB|       7|11.428571428571429|
|     Seagate|         ST4000DM000|              4TB|     269|23.795180722891565|
|         WDC| WDC WUH721414ALE6L4|             14TB|       1|              40.0|
|         WDC|      WDC WD5000LPVX|            500GB|      10|               0.0|
|HGST/Hitachi|HGST HUH721212ALN604|             12TB|      50|              3.84|
+------------+--------------------+-----------------+--------+------------------+
only showing top 5 rows

failure_bymodel: org.apache.spark.sql.DataFrame = [manufacturer: string, model: string ... 3 more fiel
ds]
```

Took 4 sec. Last updated by admin at May 04 2021, 7:01:09 AM.

```
val total_datacount_bymodel = spark.sql("SELECT manufacturer, model, capacity_bytes_TB, count(*) AS co
                                "FROM query_data "+
                                "WHERE date = '2020-12-31'"+
                                "GROUP BY model, manufacturer, capacity_bytes_TB")
total_datacount_bymodel.show(5)
```

FINISHED

```
+------------+--------------------+-----------------+-----+
|manufacturer|               model|capacity_bytes_TB|count|
+------------+--------------------+-----------------+-----+
|HGST/Hitachi|HGST HUH721212ALE600|             12TB| 2600|
|     Seagate|         ST4000DM000|              4TB|18912|
|         WDC| WDC WUH721414ALE6L4|             14TB| 5903|
|         WDC|      WDC WD5000LPVX|            500GB|  198|
|    TOSHIBHA|TOSHIBA MG07ACA14TEY|             14TB|  160|
+------------+--------------------+-----------------+-----+
only showing top 5 rows
```

```
total_datacount_bymodel: org.apache.spark.sql.DataFrame = [manufacturer: string, model: string ... 2 m
ore fields]
```

Took 1 sec. Last updated by admin at May 04 2021, 7:01:16 AM.

1

```
total_datacount_bymodel.registerTempTable("model_count")
failure_bymodel.registerTempTable("model_failures")
drivedays.registerTempTable("drivedays")
```

FINISHED

```
warning: there were three deprecation warnings; re-run with -deprecation for details
```

Took 0 sec. Last updated by admin at May 04 2021, 7:01:19 AM.

```
//since we are considering 2020 data, Annual Failure Rate can be calculated using 366 days instead of

val failure_rates = spark.sql("SELECT drivedays.manufacturer AS manufacturer, drivedays.model AS model
                              "drivedays.capacity_bytes_TB AS capacity_bytes, "+
                              "drivedays.drive_days AS drivedays, "+
                              "model_failures.failures AS failures, "+
                              "model_failures.temperature AS temperature, "+
                              "100.0 * (1.0 * failures) / (drive_days / 366.0) AS annual_failure_rat
                              "FROM drivedays, model_failures, model_count "+
                              "WHERE drivedays.model = model_failures.model "+
                              "AND model_count.model = model_failures.model "+
                              "ORDER BY model")

failure_rates.show(5)
```

FINISHED

```
+------------+--------------------+--------------+---------+--------+-----------+-------------------+
|manufacturer|               model|capacity_bytes|drivedays|failures|temperature|annual_failure_rate|
+------------+--------------------+--------------+---------+--------+-----------+-------------------+
|HGST/Hitachi|HGST HDS5C4040ALE630|           4TB|     9276|       1|        0.0|           3.945666|
|HGST/Hitachi|HGST HMS5C4040ALE640|           4TB|  1083641|       8|       19.5|           0.270200|
|HGST/Hitachi|HGST HMS5C4040ALE640|  not_detected|      133|       8|       19.5|        2201.503627|
|HGST/Hitachi|HGST HMS5C4040ALE640|  not_detected|      133|       8|       19.5|        2201.503627|
|HGST/Hitachi|HGST HMS5C4040ALE640|           4TB|  1083641|       8|       19.5|           0.270200|
+------------+--------------------+--------------+---------+--------+-----------+-------------------+
only showing top 5 rows

failure_rates: org.apache.spark.sql.DataFrame = [manufacturer: string, model: string ... 5 more field
s]
```

Took 17 sec. Last updated by admin at May 04 2021, 7:01:39 AM.

```
failure_rates.registerTempTable("Annual_failure_rate")
```

FINISHED

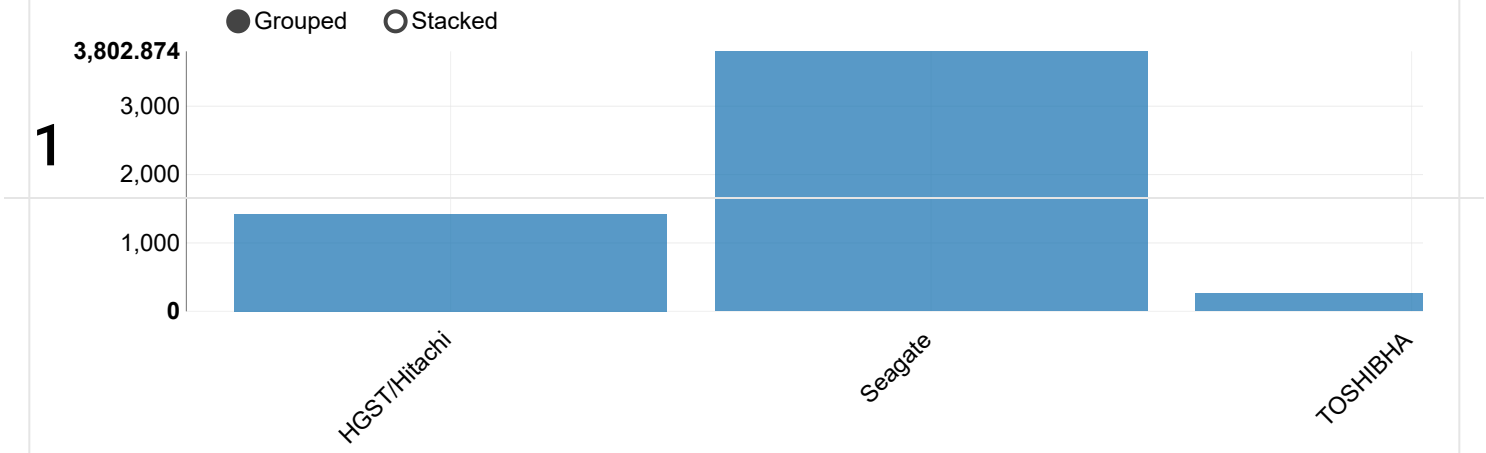
```
warning: there was one deprecation warning; re-run with -deprecation for details
```

Took 0 sec. Last updated by admin at May 04 2021, 7:01:44 AM.

```
%sql
select * from Annual_failure_rate
```

FINISHED

settings



Took 13 sec. Last updated by admin at May 04 2021, 7:02:01 AM.

READY

```
//Harddrive failure prediction
//Back blaze consider 5  smart values inorder to predict the drive failure. If one of these values is
//Since it is very important to protect the data, failure prediction gives a better insight of when to
//Here each of those 5 smart raw values are being analysed seperately to see whether the failure of dr
```

FINISHED

```
//Smart 5
val Reallocated_Sector_Count = spark.sql("SELECT model, smart_5_raw, failure "+
                                "FROM query_data "+
                                "WHERE smart_5_raw >=1 "+
                                "group by model, smart_5_raw,failure")
Reallocated_Sector_Count.show(5)
```

```
+-------------+-----------+-------+
|        model|smart_5_raw|failure|
+-------------+-----------+-------+
|ST12000NM0007|      150.0|      0|
|ST12000NM0007|     1688.0|      0|
|ST12000NM0007|    23488.0|      0|
|  ST8000DM002|    58064.0|      0|
|  ST8000DM002|     2256.0|      0|
+-------------+-----------+-------+
only showing top 5 rows

Reallocated_Sector_Count: org.apache.spark.sql.DataFrame = [model: string, smart_5_raw: string ... 1 m
ore field]
```

Took 6 sec. Last updated by admin at May 04 2021, 7:02:12 AM.

FINISHED

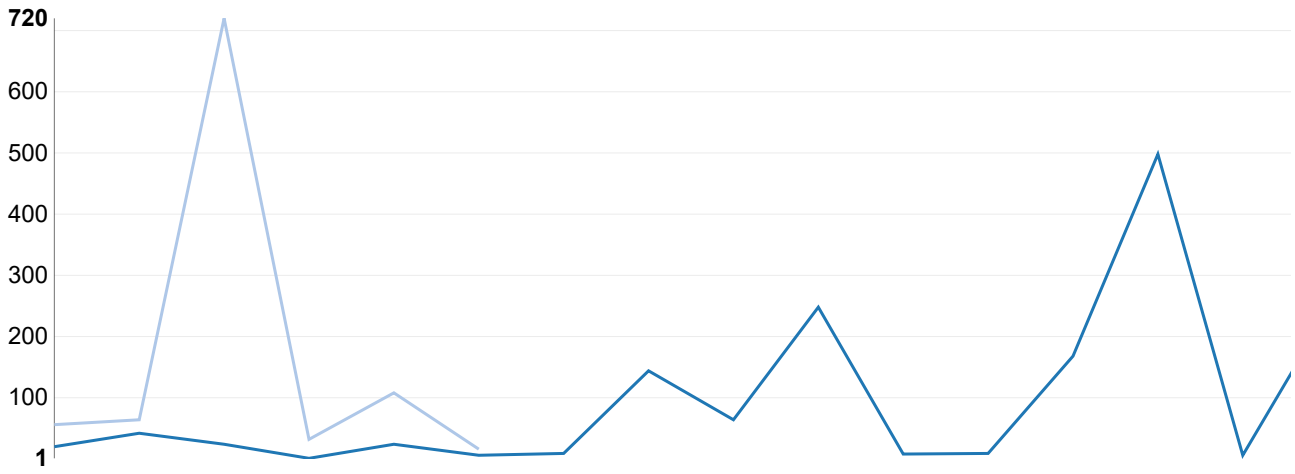
```
Reallocated_Sector_Count.createOrReplaceTempView("smart_5_data")
```

Took 0 sec. Last updated by admin at May 04 2021, 7:02:16 AM.

FINISHED

```
%sql
select * from smart_5_data
```

settings



Output is truncated to 1000 rows. Learn more about **zeppelin.spark.maxResult**

Took 5 sec. Last updated by admin at May 04 2021, 7:02:24 AM.

FINISHED

```
val Reported_Uncorrectable_Errors = spark.sql("SELECT model, smart_187_raw, failure "+
                                "FROM query_data "+
                                "WHERE smart_187_raw >=1 ")

Reported_Uncorrectable_Errors.show(5)
```

```
+-------------+-------------+-------+
|        model|smart_187_raw|failure|
+-------------+-------------+-------+
|  ST4000DM000|         29.0|      0|
|  ST4000DM000|          1.0|      0|
| ST8000NM0055|          3.0|      0|
|  ST4000DM000|          2.0|      0|
|ST12000NM0007|         22.0|      0|
+-------------+-------------+-------+
only showing top 5 rows

Reported_Uncorrectable_Errors: org.apache.spark.sql.DataFrame = [model: string, smart_187_raw: string
... 1 more field]
```

Took 1 sec. Last updated by admin at May 04 2021, 7:02:29 AM.

```
Reported_Uncorrectable_Errors.createOrReplaceTempView("smart_187_data")
```

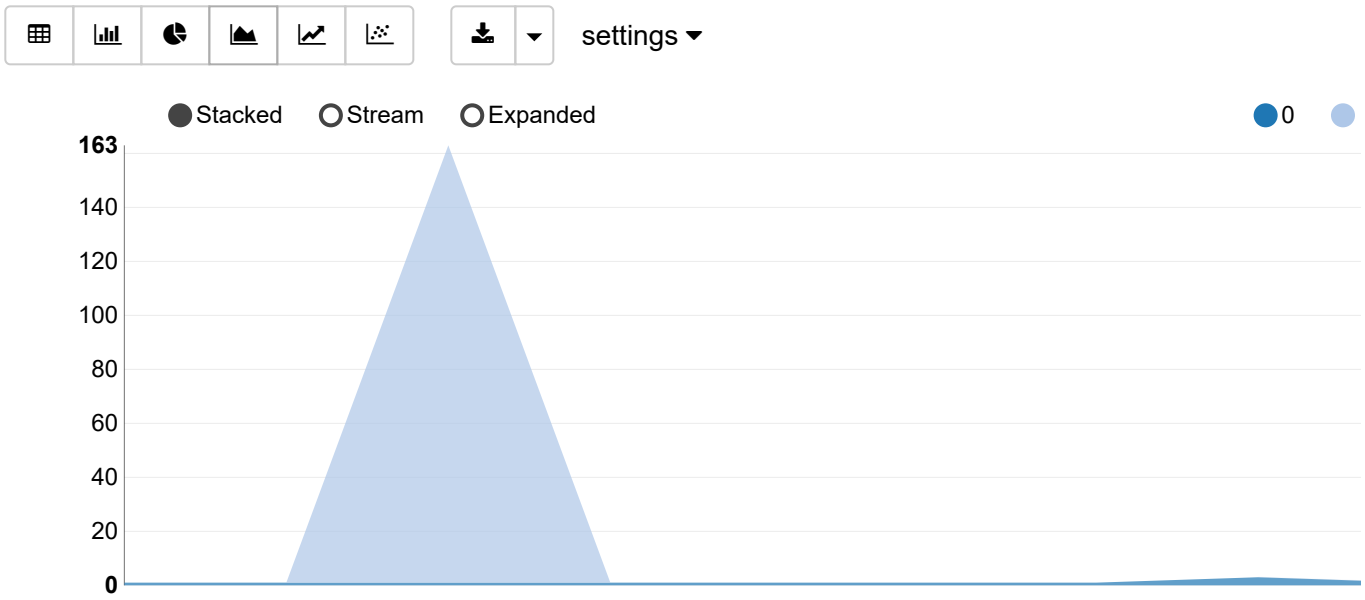
FINISHED

Took 0 sec. Last updated by admin at May 04 2021, 7:02:32 AM.

1

```
%sql
select * from smart_187_data
```

FINISHED



Output is truncated to 1000 rows. Learn more about **zeppelin.spark.maxResult**

Took 0 sec. Last updated by admin at May 04 2021, 7:02:36 AM.

```
val Command_Timeout = spark.sql("SELECT model, smart_188_raw, failure "+
                                "FROM query_data "+
                                "WHERE smart_188_raw >=1 ")

Command_Timeout.show(5)
```

FINISHED

```
+-------------+-------------+-------+
|        model|smart_188_raw|failure|
+-------------+-------------+-------+
|  ST8000DM002|          2.0|      0|
|  ST8000DM002|          8.0|      0|
|MTFDDAV240TDU|         28.0|      0|
|ST12000NM0007|          1.0|      0|
|ST12000NM0007|      65541.0|      0|
+-------------+-------------+-------+
only showing top 5 rows

Command_Timeout: org.apache.spark.sql.DataFrame = [model: string, smart_188_raw: string ... 1 more fie
ld]
```

Took 0 sec. Last updated by admin at May 04 2021, 7:02:41 AM.

```
Command_Timeout.createOrReplaceTempView("smart_188_data")
```

FINISHED

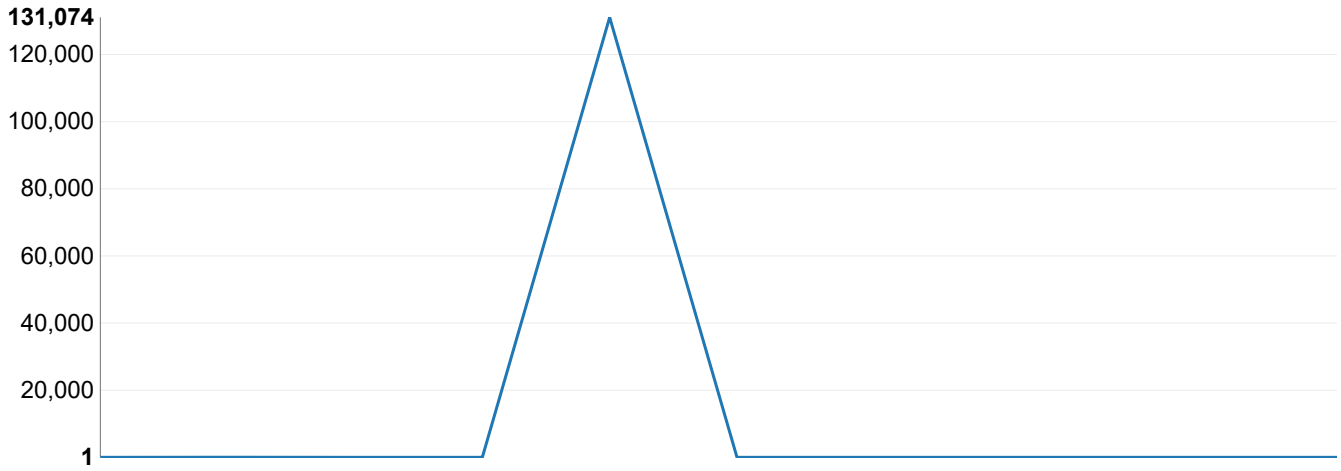
Took 0 sec. Last updated by admin at May 04 2021, 7:02:45 AM.

1

```
%sql
select * from smart_188_data
```

FINISHED

settings



Output is truncated to 1000 rows. Learn more about **zeppelin.spark.maxResult**

Took 0 sec. Last updated by admin at May 04 2021, 7:02:49 AM.

```
val Current_Pending_Sector_Count = spark.sql("SELECT model, smart_197_raw, failure "+
                                "FROM query_data "+
                                "WHERE smart_197_raw >=1 ")

Current_Pending_Sector_Count.show(5)
```

FINISHED

```
+-------------+-------------+-------+
|        model|smart_197_raw|failure|
+-------------+-------------+-------+
|  ST4000DM000|         64.0|      0|
| ST8000NM0055|          8.0|      0|
|  ST4000DM000|          8.0|      0|
|  ST4000DM000|         32.0|      0|
|ST10000NM0086|          8.0|      0|
+-------------+-------------+-------+
only showing top 5 rows

Current_Pending_Sector_Count: org.apache.spark.sql.DataFrame = [model: string, smart_197_raw: string
... 1 more field]
```

Took 0 sec. Last updated by admin at May 04 2021, 7:02:53 AM.

```
Current_Pending_Sector_Count.createOrReplaceTempView("smart_197_data")
```

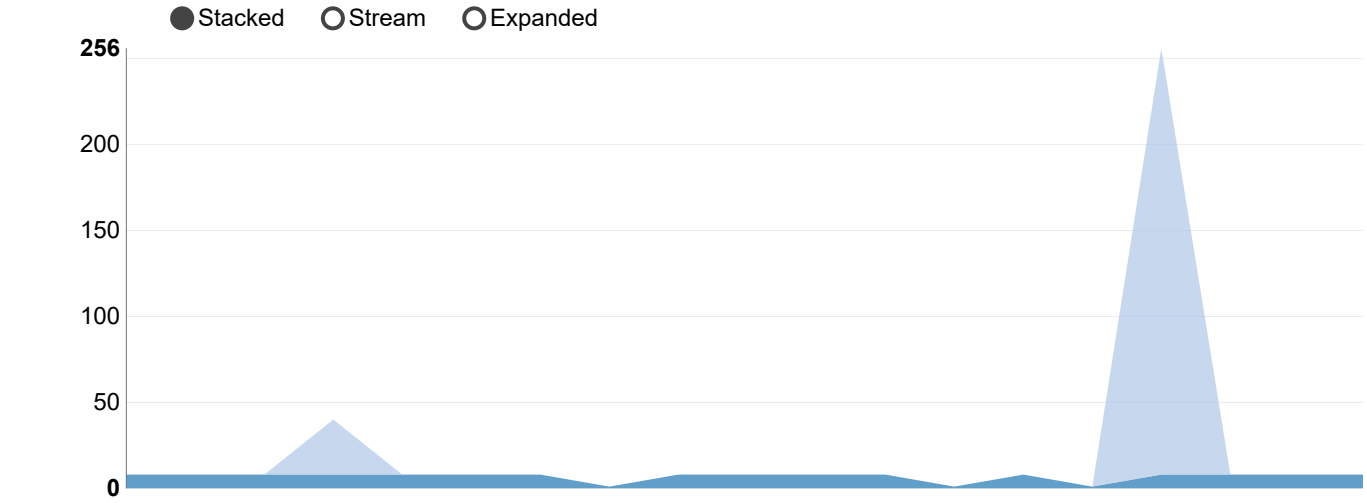
FINISHED

Took 1 sec. Last updated by admin at May 04 2021, 7:02:57 AM.

```
%sql
select * from smart_197_data
```

FINISHED

settings



Output is truncated to 1000 rows. Learn more about **zeppelin.spark.maxResult**

Took 0 sec. Last updated by admin at May 04 2021, 7:03:00 AM.

```
val Offline_Uncorrectable = spark.sql("SELECT model, smart_198_raw, failure "+
                                "FROM query_data "+
                                "WHERE smart_198_raw >= 100 ")


Offline_Uncorrectable.show(5)
```

FINISHED

```
+-------------+-------------+-------+
|        model|smart_198_raw|failure|
+-------------+-------------+-------+
| ST8000NM0055|        128.0|      0|
|ST10000NM0086|        112.0|      0|
|  ST4000DM000|        488.0|      0|
|  ST4000DM000|      42200.0|      0|
|ST12000NM0007|        744.0|      0|
+-------------+-------------+-------+
only showing top 5 rows
```

```
Offline_Uncorrectable: org.apache.spark.sql.DataFrame = [model: string, smart_198_raw: string ... 1 mo
re field]
```

Took 0 sec. Last updated by admin at May 04 2021, 7:03:03 AM.

1

```
Offline_Uncorrectable.createOrReplaceTempView("smart_198_data")
```

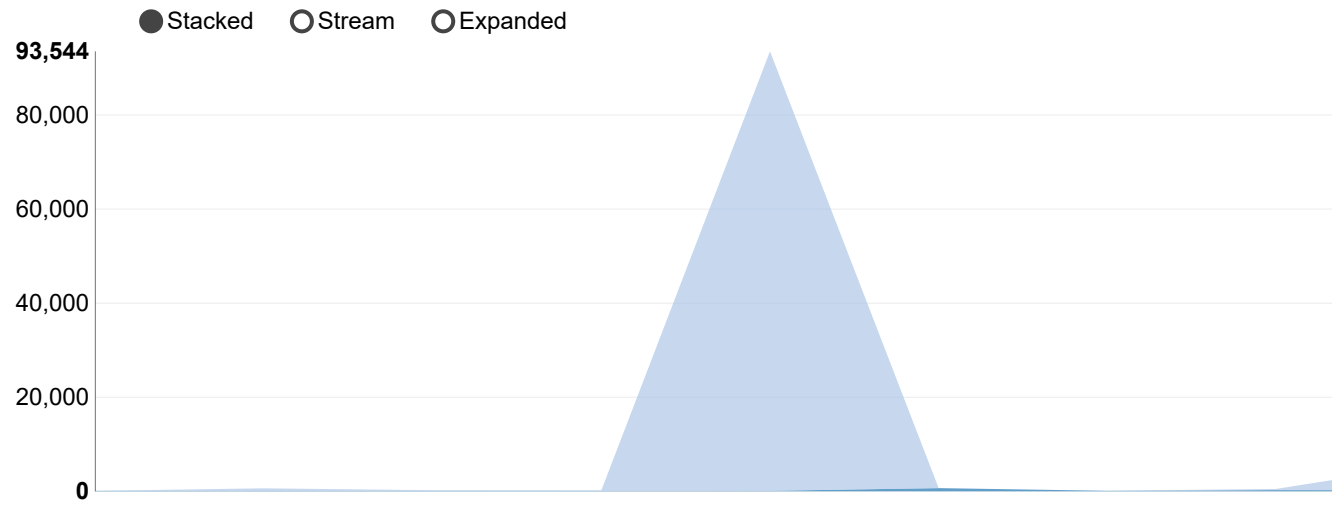
FINISHED

Took 1 sec. Last updated by admin at May 04 2021, 7:03:07 AM.

```
%sql
select * from smart_198_data
```

FINISHED

settings



Output is truncated to 1000 rows. Learn more about **zeppelin.spark.maxResult**

Took 1 sec. Last updated by admin at May 04 2021, 7:03:11 AM.

READY