CASE STUDY 3 **WORKING WITH SENSOR DATA**

In this case study, there are two datasets-

1. **building.csv** contains the details of the top 20 buildings all over the world and
2. **HVAC.csv** contains the target temperature and the actual temperature along with the BuildingID.

HVAC (heating, ventilating/ventilation, and air conditioning) is the technology of indoor and vehicular environmental comfort. Its goal is to provide thermal comfort and acceptable indoor air quality. Through the HVAC sensors, we will get the temperature of the buildings.

Details of the both the tables are as follows:

Building.csv –BuildingID, BuildingMgr, BuildingAge, HVACproduct, Country
HVAC.csv –Date, Time, TargetTemp, ActualTemp, System, SystemAge, BuildingID.

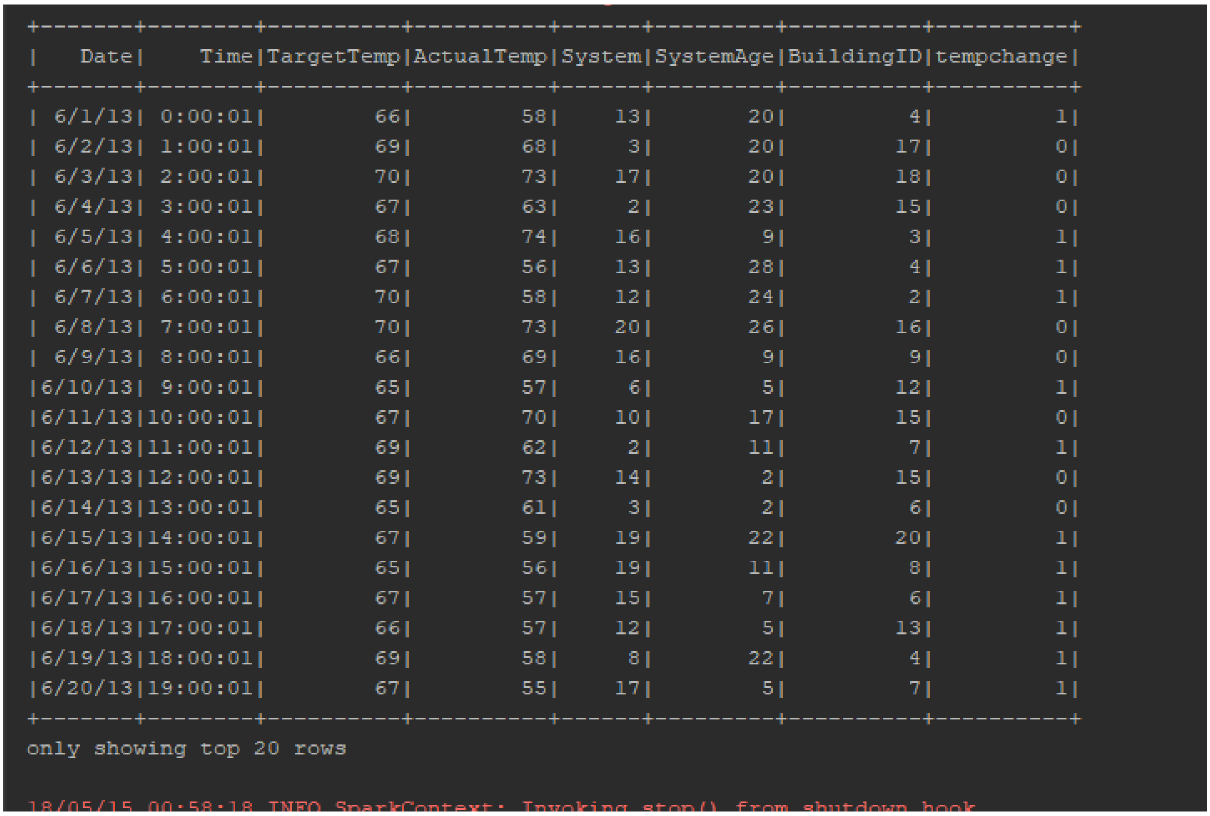
We have the following objectives to perform on the above data sets, which we will perform the in **IntelliJ IDEA** application to get the results

Objective-1

- Load HVAC.csv file into temporary table.
- Add a new column, tempchange -set to 1, if there is a change of greater than +/-5 between actual and target temperature.

```
package Sensor_Data_Analysis

import org.apache.spark.sql.SparkSession

object Objective1

{

  def main(args: Array[String]): Unit =

{

  println("Sensor data analysis!!!")


 // Use new SparkSession interface in Spark

val spark = SparkSession.builder().master("local").appName("Working with Sensor Data")
.config("spark.some.config.option", "some-value").getOrCreate()


// load the dataset using the csvFile method

val hvac_data = spark.read.format("com.databricks.spark.csv").option("header","true")
.option("inferSchema","true")
.load("C:\\Users\\Bhaskar\\Desktop\\AcadGild\\CaseStudies\\CaseStudy3SensorCaseStudy\\Dataset\\HVAC.csv")


  //convert the hvac RDD into dataframe

   val hvac_data_df = hvac_data.toDF


//Register or load hvac dataframe into temporary table ‘hvacTemptable’

   hvac_data_df.registerTempTable("hvacTempTable")


   println("hvac dataframe is loaded in  hvacTempTable !")


//use spark sql query to add extra column tempchange in hvacTempTable

val hvac_temp_chnage = spark.sql("select *,IF((targettemp - actualtemp) > 5, '1',
                                                    IF((targettemp - actualtemp) < -5, '1', 0)) AS tempchange
                                                     from hvacTempTable")

             hvac_temp_chnage.show()

        } }
```

Below screen shot shows the ‘**hvacTempTable**’ with extra column added which shows the value 1 for temp-difference is equal to -/+5 and 0 for difference less than -/+5



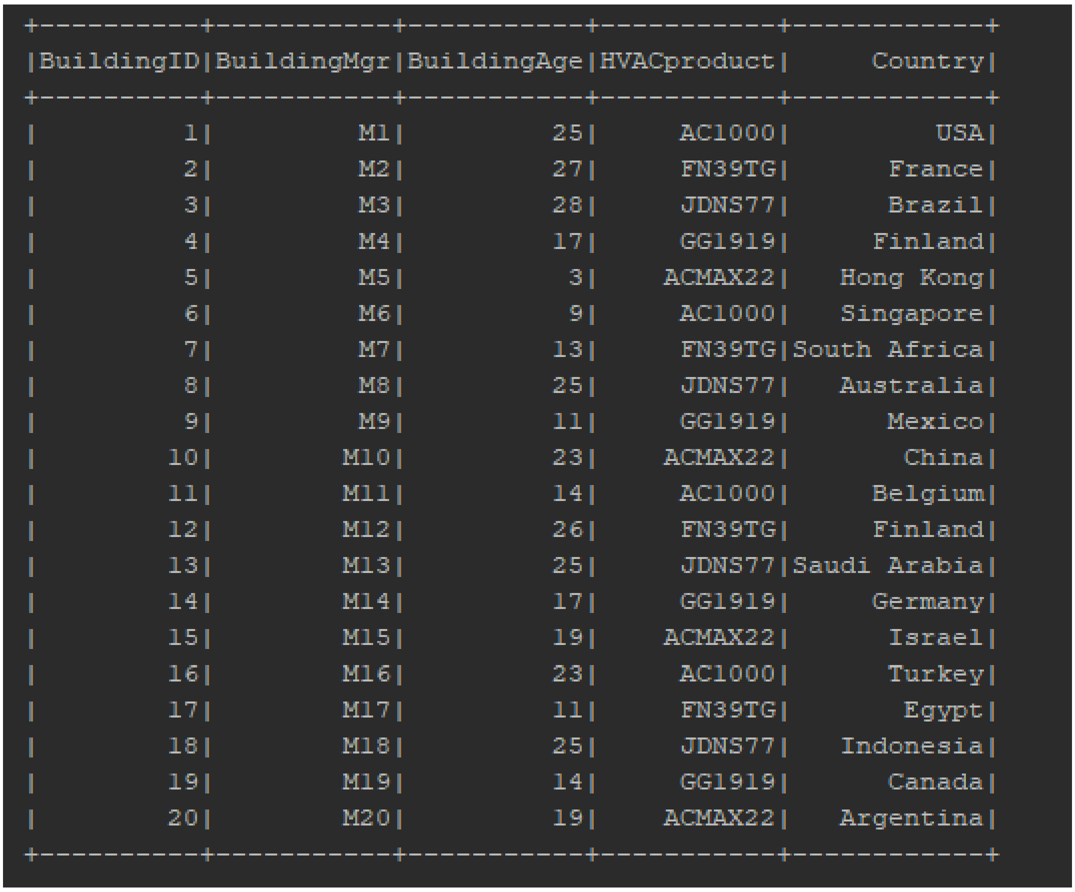
```
+-------+--------+----------+----------+------+---------+----------+----------+
|   Date|    Time|TargetTemp|ActualTemp|System|SystemAge|BuildingID|tempchange|
+-------+--------+----------+----------+------+---------+----------+----------+
| 6/1/13| 0:00:01|        66|        58|    13|       20|         4|         1|
| 6/2/13| 1:00:01|        69|        68|     3|       20|        17|         0|
| 6/3/13| 2:00:01|        70|        73|    17|       20|        18|         0|
| 6/4/13| 3:00:01|        67|        63|     2|       23|        15|         0|
| 6/5/13| 4:00:01|        68|        74|    16|        9|         3|         1|
| 6/6/13| 5:00:01|        67|        56|    13|       28|         4|         1|
| 6/7/13| 6:00:01|        70|        58|    12|       24|         2|         1|
| 6/8/13| 7:00:01|        70|        73|    20|       26|        16|         0|
| 6/9/13| 8:00:01|        66|        69|    16|        9|         9|         0|
|6/10/13| 9:00:01|        65|        57|     6|        5|        12|         1|
|6/11/13|10:00:01|        67|        70|    10|       17|        15|         0|
|6/12/13|11:00:01|        69|        62|     2|       11|         7|         1|
|6/13/13|12:00:01|        69|        73|    14|        2|        15|         0|
|6/14/13|13:00:01|        65|        61|     3|        2|         6|         0|
|6/15/13|14:00:01|        67|        59|    19|       22|        20|         1|
|6/16/13|15:00:01|        65|        56|    19|       11|         8|         1|
|6/17/13|16:00:01|        67|        57|    15|        7|         6|         1|
|6/18/13|17:00:01|        66|        57|    12|        5|        13|         1|
|6/19/13|18:00:01|        69|        58|     8|       22|         4|         1|
|6/20/13|19:00:01|        67|        55|    17|        5|         7|         1|
+-------+--------+----------+----------+------+---------+----------+----------+
only showing top 20 rows

18/05/15 00:58:18 INFO SparkContext: Invoking stop() from shutdown hook
```

Objective-2

- Load building.csv file into temporary table

```
package Sensor_Data_Analysis

import org.apache.spark.sql.SparkSession

object Objective2

{

    def main(args: Array[String]): Unit =

 {

     println("Sensor data analysis!!!")


// Use new SparkSession interface in Spark

 val spark = SparkSession .builder() .master("local").appName("Working with Sensor Data")
.config("spark.some.config.option", "some-value") .getOrCreate()


// load the dataset using the csvFile method

val building_data = spark .read.format("com.databricks.spark.csv") .option("header","true")
.option("inferSchema","true")
.load("C:\\Users\\Bhaskar\\Desktop\\AcadGild\\CaseStudies\\CaseStudy3SensorCaseStudy\\Dataset\\building
.csv")


  //convert the hvac RDD into dataframe

val building_data_df =building_data.toDF


//Register or load hvac dataframe into temporary table ‘hvacTemptable’

building_data_df.registerTempTable("buildingTempTable")

//use spark sql to show the loaded building data in buildingTempTable

val load = spark.sql("select * from buildingTempTable").show()

  }}
```

Below screen shots shows the 20 rows of building data-set from ‘**bulidingTempTable**’



```
+----------+-----------+-----------+----------+------------+
|BuildingID|BuildingMgr|BuildingAge|HVACproduct|     Country|
+----------+-----------+-----------+----------+------------+
|         1|         M1|         25|    AC1000|         USA|
|         2|         M2|         27|    FN39TG|      France|
|         3|         M3|         28|    JDNS77|      Brazil|
|         4|         M4|         17|    GG1919|     Finland|
|         5|         M5|          3|   ACMAX22|   Hong Kong|
|         6|         M6|          9|    AC1000|   Singapore|
|         7|         M7|         13|    FN39TG|South Africa|
|         8|         M8|         25|    JDNS77|   Australia|
|         9|         M9|         11|    GG1919|      Mexico|
|        10|        M10|         23|   ACMAX22|       China|
|        11|        M11|         14|    AC1000|     Belgium|
|        12|        M12|         26|    FN39TG|     Finland|
|        13|        M13|         25|    JDNS77|Saudi Arabia|
|        14|        M14|         17|    GG1919|     Germany|
|        15|        M15|         19|   ACMAX22|      Israel|
|        16|        M16|         23|    AC1000|      Turkey|
|        17|        M17|         11|    FN39TG|       Egypt|
|        18|        M18|         25|    JDNS77|   Indonesia|
|        19|        M19|         14|    GG1919|      Canada|
|        20|        M20|         19|   ACMAX22|   Argentina|
+----------+-----------+-----------+----------+------------+
```

Objective-3

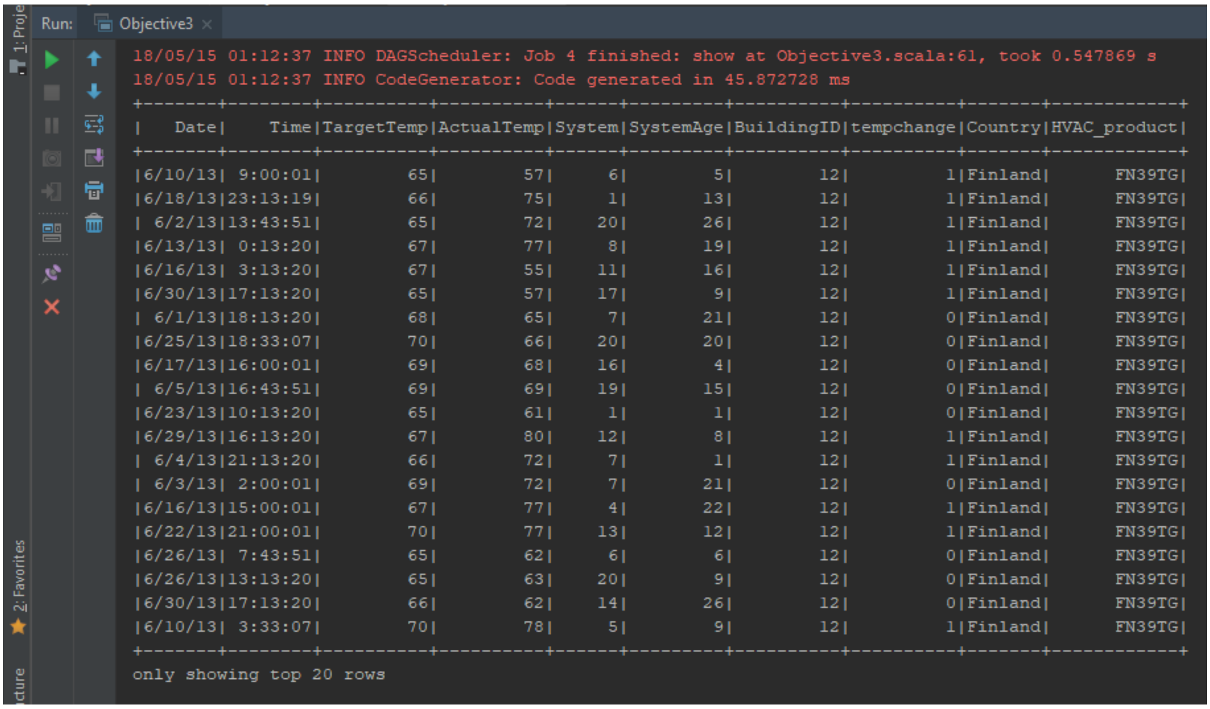
- Figure out the number of times, temperature has changed by 5 degrees or more for each country.

```
package Sensor_Data_Analysis

import org.apache.spark.sql.SparkSession

object Objective3

 {

//declare a case class HVAC holding the dataset description of the hvac-data.

case class
HVAC(Date:String,Time:String,TargetTemp:Int,ActualTemp:Int,System:Int,SystemAge:Int,BuildingID:Int)

//declare a case class BHUILDING holding the dataset description of the building-data.

case class BUILDING(BuildingID:Int,BuildingMgr:String,BuildingAge:Int,HVAC_Product:String,Country:String)


  def main(args: Array[String]): Unit =
  {
    println("Sensor data analysis!!!")
//Use new SparkSession interface in spark
    val spark = SparkSession .builder().master("local") .appName("Working with Sensor Data")
                    .config("spark.some.config.option", "some-value") .getOrCreate()
      println("Spark Session is created !!!")


//load the hvac dataset using the textFile method
val hvac_data_with_header = spark.sparkContext.textFile("C:\\Users\\Bhaskar\\Desktop\\AcadGild\\CaseStudies\\CaseStudy3SensorCaseStudy\\Dataset\\HVAC.csv")


//creating a variable header, which holds the first line of the dataset, in our data set hvac.csv the first line is a header line.
val header = hvac_data_with_header.first()


//filter the header line from the dataset using the filter RDD
val hvac_data = hvac_data_with_header.filter(row => row != header)


//For implicit conversions like converting RDDs and sequences  to DataFrames
    import spark.implicits._


//preparing a structure for the data, mapping it to the case class structure, and finally converting it to a data frame.
    val hvac_data_df = hvac_data.map(x=>x.split(",")).map(x => HVAC(x(0),x(1),x(2).toInt,x(3).toInt,x(4).toInt,x(5).toInt,x(6).toInt)).toDF()


//Register temporary table hvacTempTable
    hvac_data_df.registerTempTable("hvacTempTable")


//Use spark-sql queru to add extra xolum for temperature change
    val hvac_1 = spark.sql("select *,IF((targettemp - actualtemp) > 5, '1', IF((targettemp - actualtemp) < -5, '1', 0)) AS tempchange from hvacTempTable")


// Register the new dataframe hvac1 into temporary table hvac1TempTable
```

```
    hvac_1.registerTempTable("hvac1TempTable")


    println("Data Frame Registered as hvac1TempTable table !")
//load the building dataset using the textFile method

 val building_data_with_header = spark.sparkContext.textFile("C:\\Users\\Bhaskar\\Desktop\\AcadGild\\CaseStudies\\CaseStudy3SensorCaseStudy\\Dataset\\building.csv")


//creating a variable header, which holds the first line of the dataset, in our data set building.csv the first line is a header line

val header1 = building_data_with_header.first()


//filter the header line from the dataset using the filter RDD

 val building_data = building_data_with_header.filter(row => row != header1)




//preparing a structure for the data, mapping it to the case class structure, and finally converting it to a data frame.

 val building_data_df = building_data.map(x=> x.split(",")).map(x =>
BUILDING(x(0).toInt,x(1),x(2).toInt,x(3),x(4))).toDF


//Register temporary table

 building_data_df.registerTempTable("buildingTempTable")


//use spark-sql query to join to tables to get columns which help to filter country and tempchange column

 val join_hvac_building = spark.sql("select h.*, b.Country, b.HVAC_product from buildingTempTable b join hvac1TempTable h on b.BuildingID = h.BuildingID")


join_hvac_building.show()
```

Below two output tables shows the details for only country **Finland** as default it select the 20 rows and randomly its select the country Finland.

<u>Below screen shot shows the join table of **hvacTempTable** and **buildingTemptable**</u>



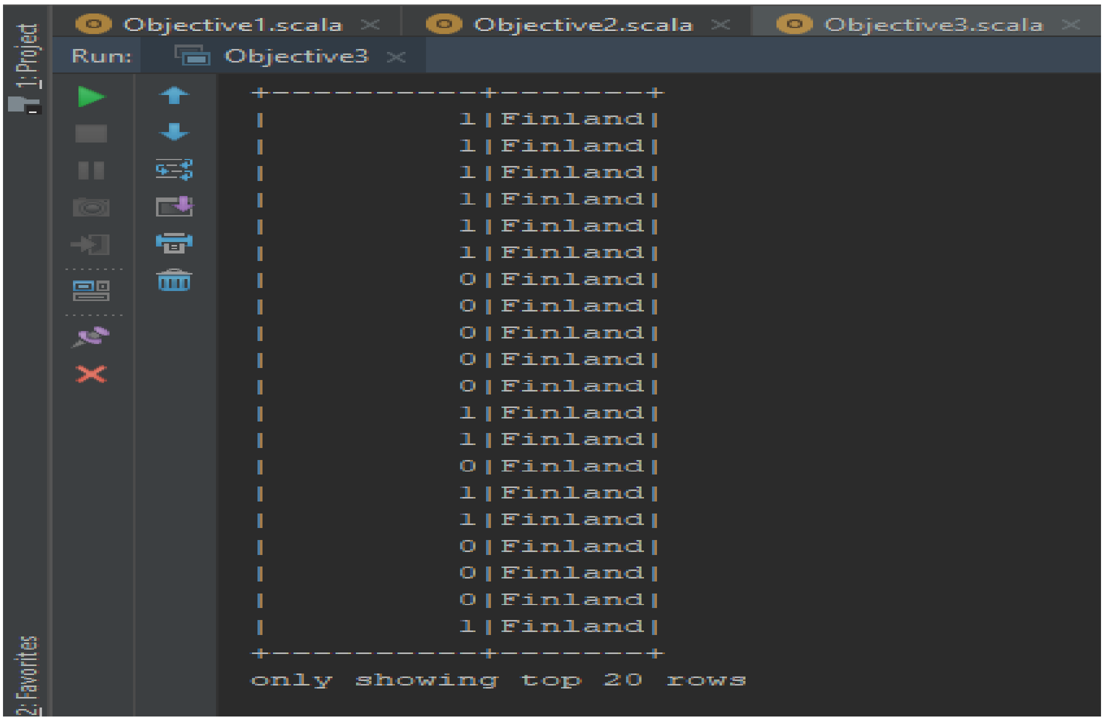
```
18/05/15 01:12:37 INFO DAGScheduler: Job 4 finished: show at Objective3.scala:61, took 0.547869 s
18/05/15 01:12:37 INFO CodeGenerator: Code generated in 45.872728 ms
+-------+--------+----------+----------+------+---------+----------+----------+-------+------------+
|   Date|    Time|TargetTemp|ActualTemp|System|SystemAge|BuildingID|tempchange|Country|HVAC_product|
+-------+--------+----------+----------+------+---------+----------+----------+-------+------------+
|6/10/13| 9:00:01|        65|        57|     6|        5|        12|         1|Finland|      FN39TG|
|6/18/13|23:13:19|        66|        75|     1|       13|        12|         1|Finland|      FN39TG|
| 6/2/13|13:43:51|        65|        72|    20|       26|        12|         1|Finland|      FN39TG|
|6/13/13| 0:13:20|        67|        77|     8|       19|        12|         1|Finland|      FN39TG|
|6/16/13| 3:13:20|        67|        55|    11|       16|        12|         1|Finland|      FN39TG|
|6/30/13|17:13:20|        65|        57|    17|        9|        12|         1|Finland|      FN39TG|
| 6/1/13|18:13:20|        68|        65|     7|       21|        12|         0|Finland|      FN39TG|
|6/25/13|18:33:07|        70|        66|    20|       20|        12|         0|Finland|      FN39TG|
|6/17/13|16:00:01|        69|        68|    16|        4|        12|         0|Finland|      FN39TG|
| 6/5/13|16:43:51|        69|        69|    19|       15|        12|         0|Finland|      FN39TG|
|6/23/13|10:13:20|        65|        61|     1|        1|        12|         0|Finland|      FN39TG|
|6/29/13|16:13:20|        67|        80|    12|        8|        12|         1|Finland|      FN39TG|
| 6/4/13|21:13:20|        66|        72|     7|        1|        12|         1|Finland|      FN39TG|
| 6/3/13| 2:00:01|        69|        72|     7|       21|        12|         0|Finland|      FN39TG|
|6/16/13|15:00:01|        67|        77|     4|       22|        12|         1|Finland|      FN39TG|
|6/22/13|21:00:01|        70|        77|    13|       12|        12|         1|Finland|      FN39TG|
|6/26/13| 7:43:51|        65|        62|     6|        6|        12|         0|Finland|      FN39TG|
|6/26/13|13:13:20|        65|        63|    20|        9|        12|         0|Finland|      FN39TG|
|6/30/13|17:13:20|        66|        62|    14|       26|        12|         0|Finland|      FN39TG|
|6/10/13| 3:33:07|        70|        78|     5|        9|        12|         1|Finland|      FN39TG|
+-------+--------+----------+----------+------+---------+----------+----------+-------+------------+
only showing top 20 rows
```

```
 //Select temperature change and country column from above
```

val tempCountry = join_hvac_building.map(x => (new Integer(x(7).toString),x(8).toString))

tempCountry.show()

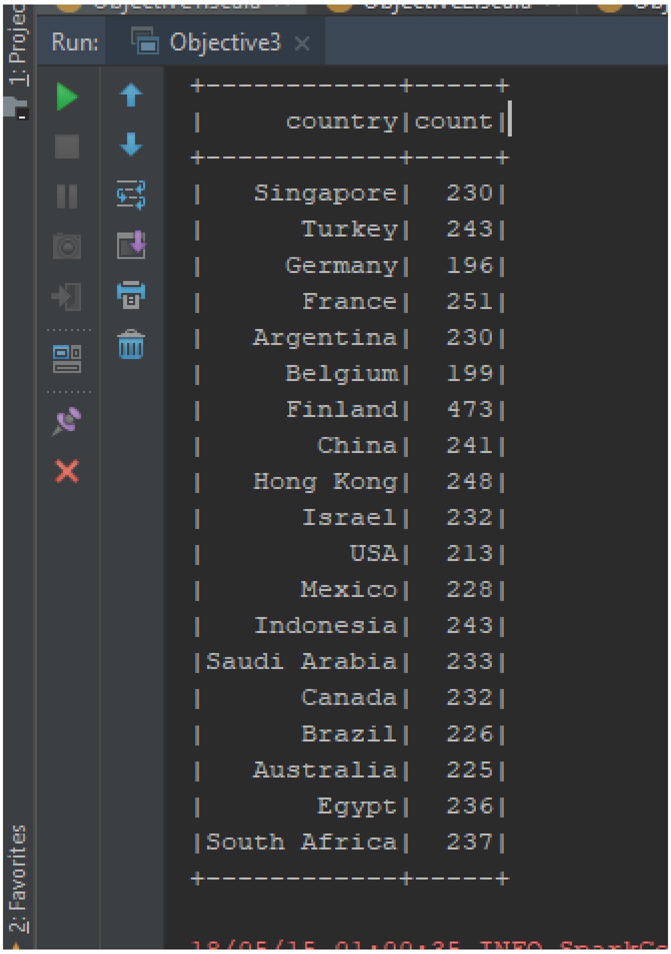
<u>Below screen shot shows the two columns **tempchange** and **country**</u>



```
+-----------+-------+
|          1|Finland|
|          1|Finland|
|          1|Finland|
|          1|Finland|
|          1|Finland|
|          1|Finland|
|          0|Finland|
|          0|Finland|
|          0|Finland|
|          0|Finland|
|          0|Finland|
|          1|Finland|
|          1|Finland|
|          0|Finland|
|          1|Finland|
|          1|Finland|
|          0|Finland|
|          0|Finland|
|          0|Finland|
|          1|Finland|
+-----------+-------+
only showing top 20 rows
```

```
  //Filter the values 1 which for temperature change greater than 5 or more

val tempCountryOnes = tempCountry.filter(x=> {if(x._1==1) true else false})


 tempCountryOnes.groupBy("_2").count.withColumnRenamed("_2","country" )show()

  }

}
```

Below screen, shot shows the number of times temperature has changed by 5 degree or more for each country.



```
+------------+-----+
|     country|count|
+------------+-----+
|   Singapore|  230|
|      Turkey|  243|
|     Germany|  196|
|      France|  251|
|   Argentina|  230|
|     Belgium|  199|
|     Finland|  473|
|       China|  241|
|   Hong Kong|  248|
|      Israel|  232|
|         USA|  213|
|      Mexico|  228|
|   Indonesia|  243|
|Saudi Arabia|  233|
|      Canada|  232|
|      Brazil|  226|
|   Australia|  225|
|       Egypt|  236|
|South Africa|  237|
+------------+-----+

18/05/15 01:00:35 INFO SparkCo
```