Case Study3:

Task:

In the given link there are two datasets; building.csv contains the details of the top 20 buildings all over the world and HVAC.csv contains the target temperature and the actual temperature along with the building Id.

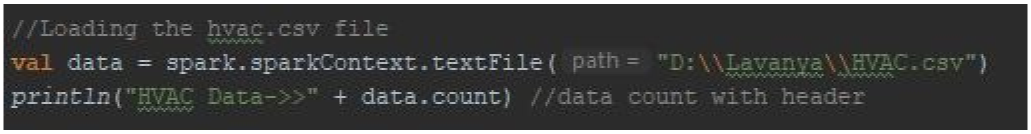
HVAC (heating, ventilating/ventilation, and air conditioning) is the technology of indoor and vehicular environmental comfort. Its goal is to provide thermal comfort and acceptable indoor air quality. Through the HVAC sensors, we will get the temperature of the buildings.

Here are the columns that are present in the datasets: Building.csv – BuildingID, BuildingMgr, BuildingAge, HVACproduct,Country HVAC.csv – Date, Time, TargetTemp, ActualTemp, System, SystemAge, BuildingID

Objective 1:

Load HVAC.csv file into temporary table

Code snapshot:

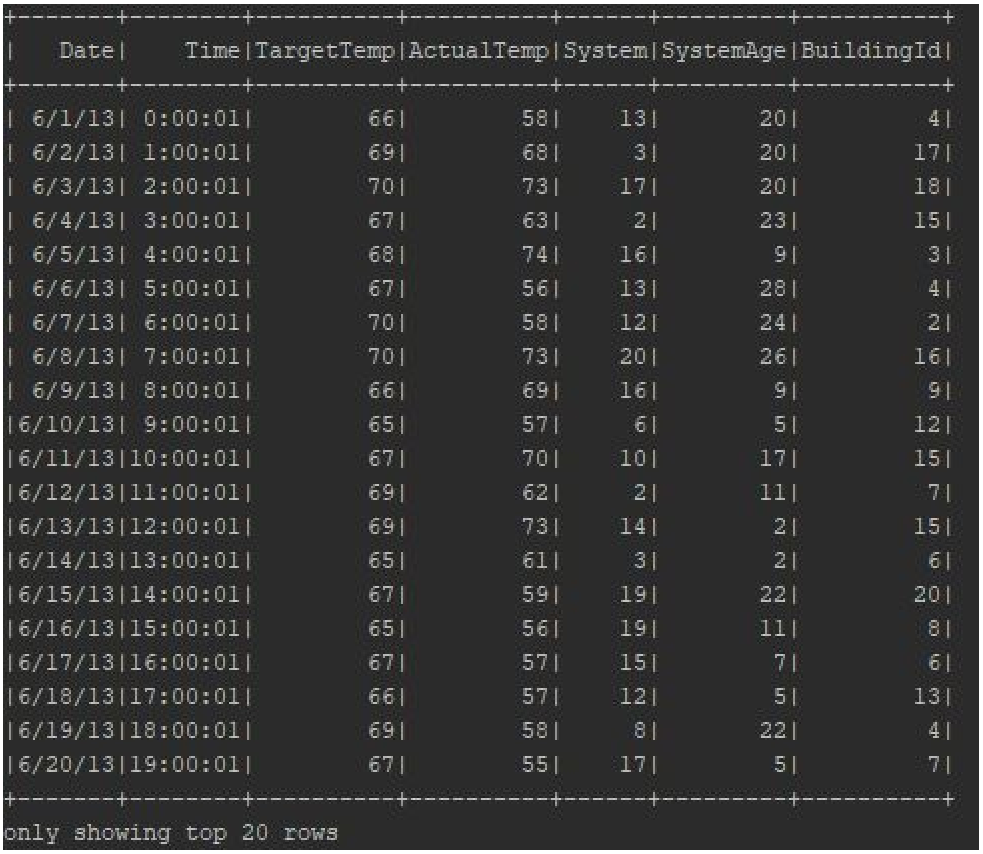


```
//Loading the hvac.csv file
val data = spark.sparkContext.textFile( path = "D:\\Lavanya\\HVAC.csv")
println("HVAC Data->>" + data.count) //data count with header
```

Data frame created after loading the file:

Showing top 20 rows of the full data frame.

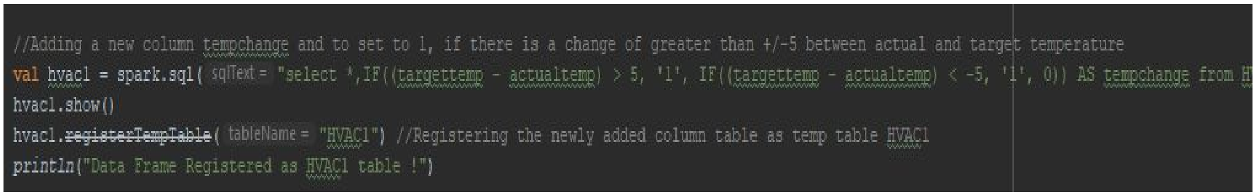
Output:



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

Add a new column, tempchange - set to 1, if there is a change of greater than +/-5 between actual and target temperature

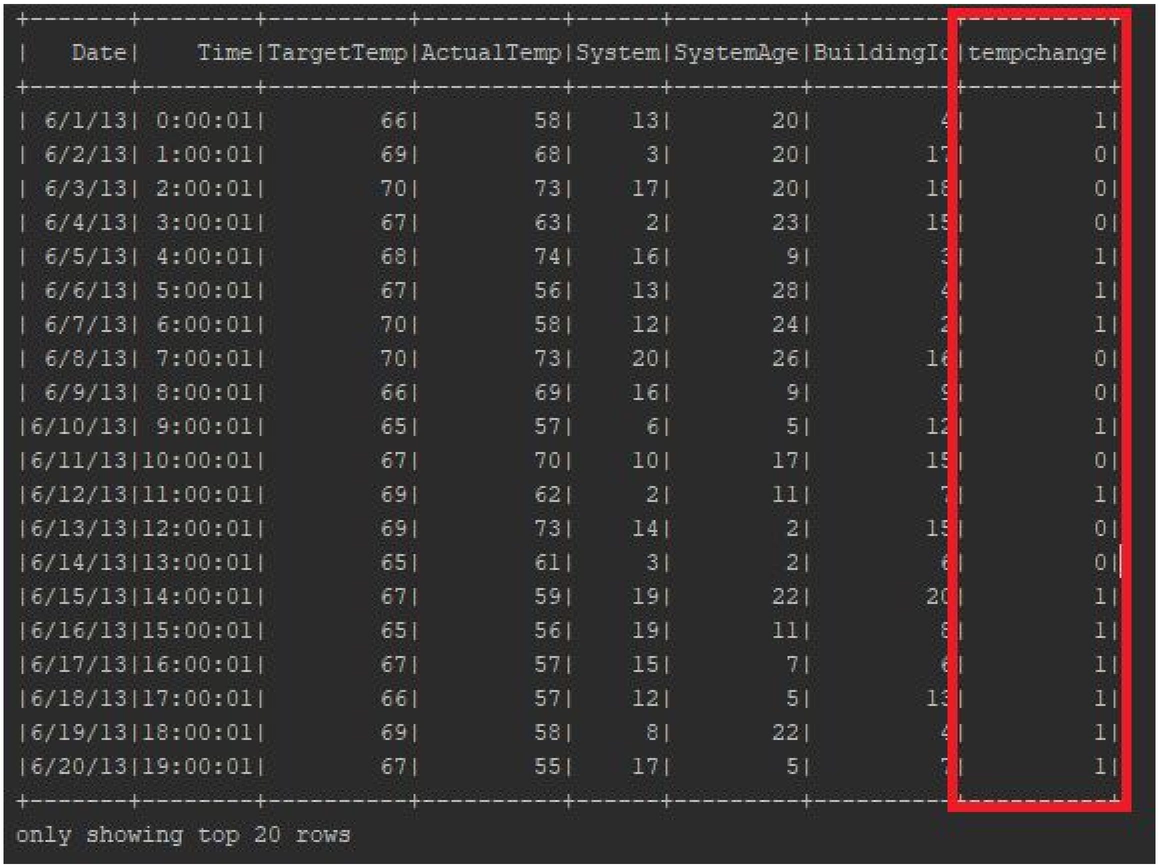
Code snapshot:



```
//Adding a new column tempchange and to set to 1, if there is a change of greater than +/-5 between actual and target temperature
val hvac1 = spark.sql( sqlText = "select *,IF((targettemp - actualtemp) > 5, '1', IF((targettemp - actualtemp) < -5, '1', 0)) AS tempchange from H
hvac1.show()
hvac1.~~registerTempTable~~( tableName = "HVAC1") //Registering the newly added column table as temp table HVAC1
println("Data Frame Registered as HVAC1 table !")
```

Output:

Highlighted in red is the newly added column.

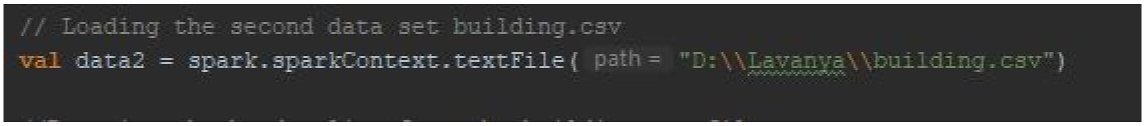


```
+--------+--------+----------+----------+------+---------+----------+----------+
|    Date|    Time|TargetTemp|ActualTemp|System|SystemAge|BuildingID|tempchange|
+--------+--------+----------+----------+------+---------+----------+----------+
|  6/1/13| 0:00:01|        66|        58|    13|       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
```

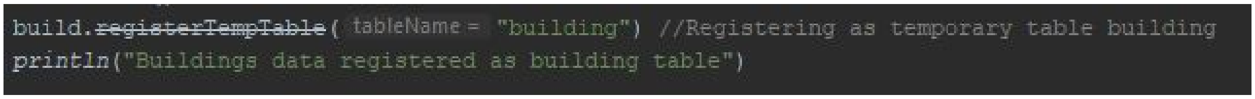
Objective 2:

Load building.csv file into temporary table

Code snapshot:



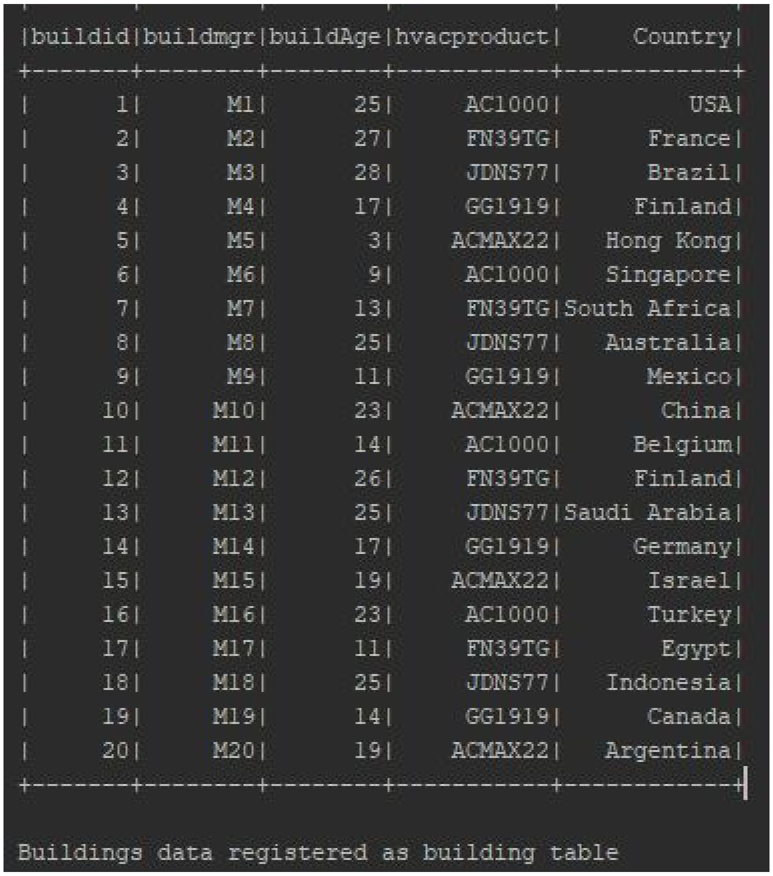
```
// Loading the second data set building.csv
val data2 = spark.sparkContext.textFile( path = "D:\\Lavanya\\building.csv")
```



```
build.~~registerTempTable~~( tableName = "building") //Registering as temporary table building
println("Buildings data registered as building table")
```

Output:

Building data frame created and registered as building table.



```
|buildid|buildmgr|buildAge|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|
+-------+--------+--------+-----------+------------+

Buildings data registered as building table
```

Objective 3:

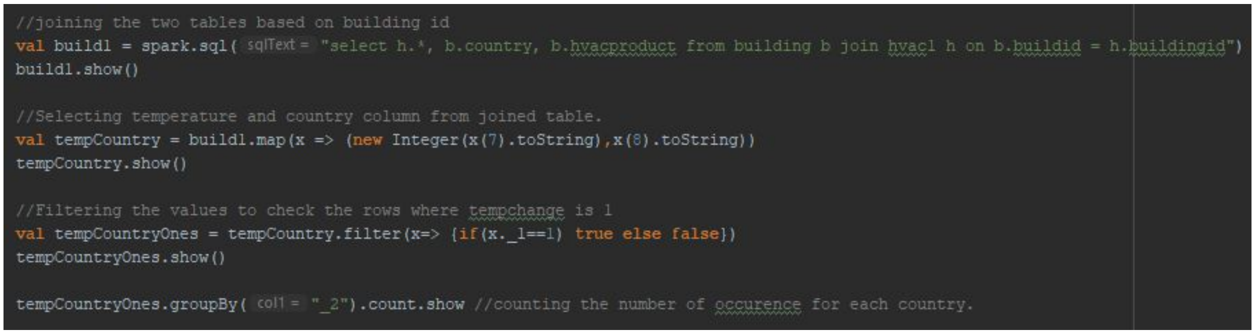
Figure out the number of times, temperature has changed by 5 degrees or more for each country:

○ Join both the tables.

○ Select tempchange and country column

○ Filter the rows where tempchange is 1 and count the number of occurrence for each country

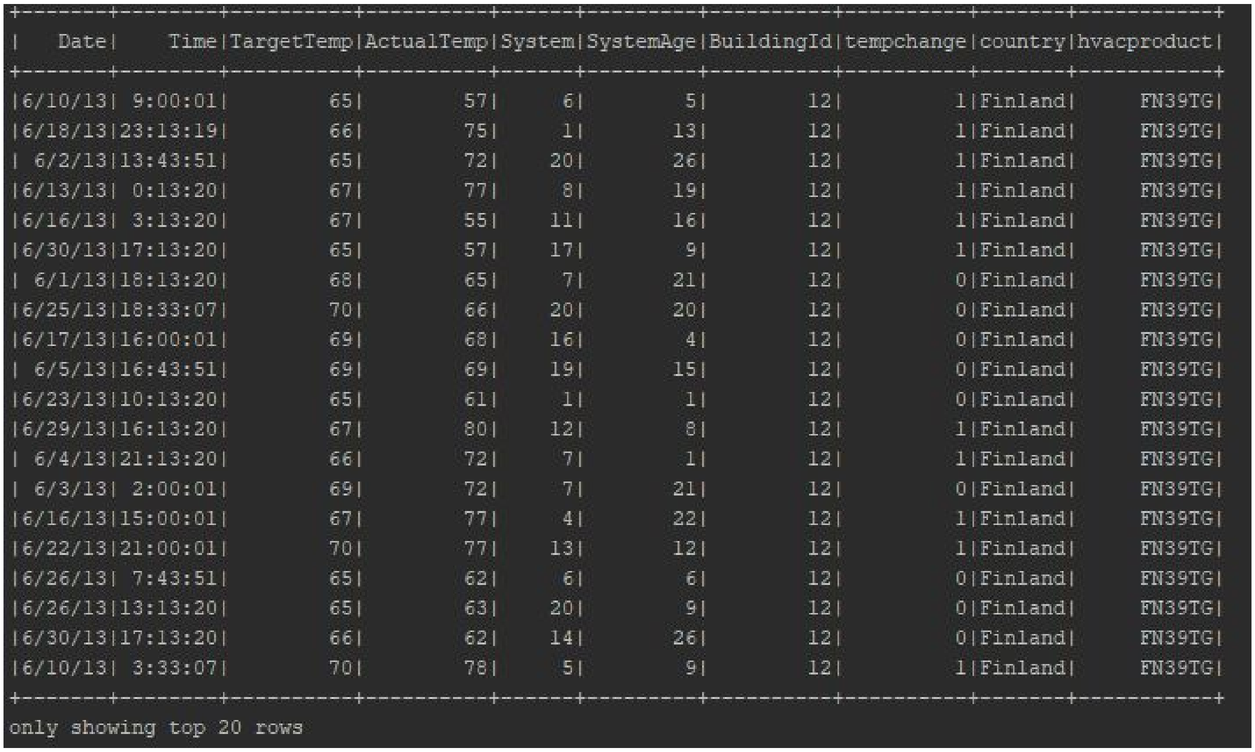
Code snapshot:



```
//joining the two tables based on building id
val buildl = spark.sql( sqlText = "select h.*, b.country, b.hvacproduct from building b join hvacl h on b.buildid = h.buildingid")
buildl.show()

//Selecting temperature and country column from joined table.
val tempCountry = buildl.map(x => (new Integer(x(7).toString),x(8).toString))
tempCountry.show()

//Filtering the values to check the rows where tempchange is 1
val tempCountryOnes = tempCountry.filter(x=> {if(x._1==1) true else false})
tempCountryOnes.show()

tempCountryOnes.groupBy( col1 = "_2").count.show //counting the number of occurence for each country.
```

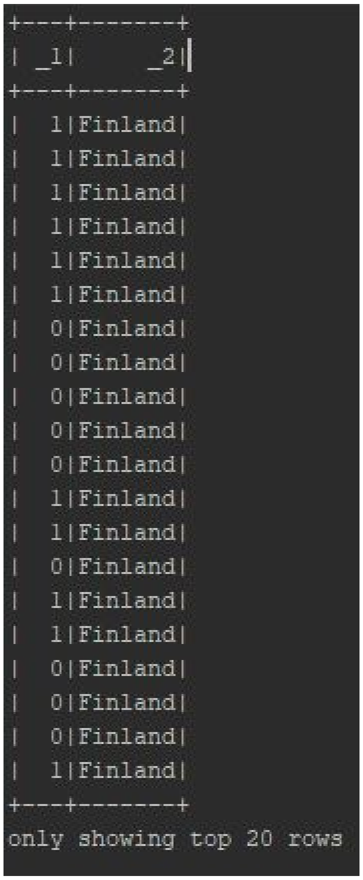
Output:

Joined table: Showing top 20 rows of the full data frame.



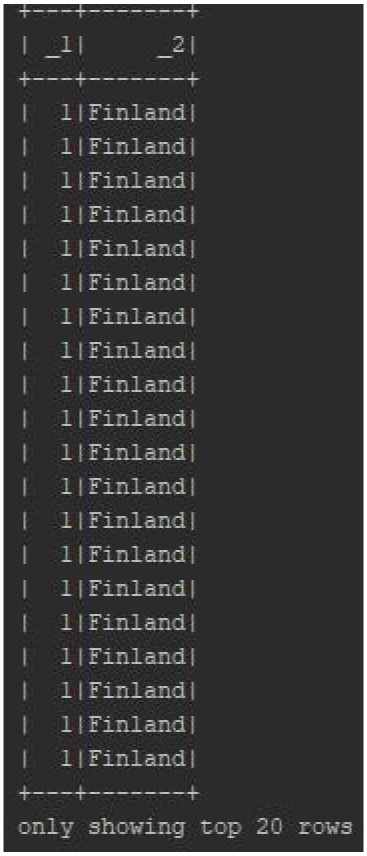
```
+-------+--------+----------+----------+------+---------+----------+----------+-------+-----------+
|   Date|    Time|TargetTemp|ActualTemp|System|SystemAge|BuildingId|tempchange|country|hvacproduct|
+-------+--------+----------+----------+------+---------+----------+----------+-------+-----------+
|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
```

Selected tempchange and country column: Showing top 20 rows of the full data frame.



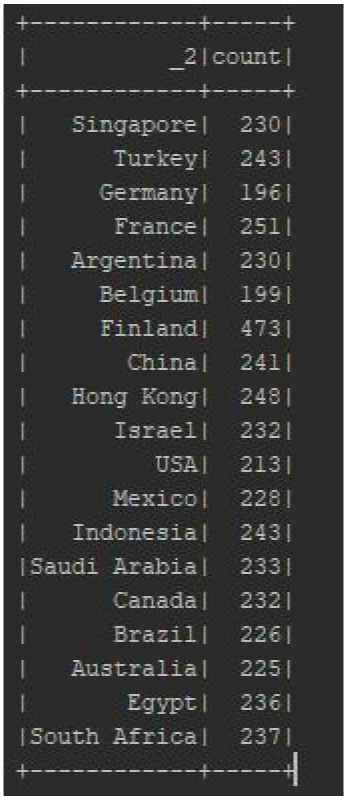
```
+---+-------+
| _1|     _2|
+---+-------+
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  0|Finland|
|  0|Finland|
|  0|Finland|
|  0|Finland|
|  0|Finland|
|  1|Finland|
|  1|Finland|
|  0|Finland|
|  1|Finland|
|  1|Finland|
|  0|Finland|
|  0|Finland|
|  0|Finland|
|  1|Finland|
+---+-------+
only showing top 20 rows
```

Filtered table containing the rows where tempchange is 1 : Showing top 20 rows of the full data frame.



```
+---+-------+
| _1|     _2|
+---+-------+
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
+---+-------+
only showing top 20 rows
```

Final Output containing number of times temperature has changed by 5 degrees or more for each country



```
+------------+-----+
|          _2|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|
+------------+-----+
```