For this data analysis, you can download the necessary dataset from this link.

In the above 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

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

Objective 2

Load building.csv file into temporary 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

Solution-:

Working with Sensor Data

In below program, we have created case classes for **building** and **hvac** data files and then created Spark object.

Scala code :

```
import org.apache.spark.sql.SparkSession

object Case_Study_3_Sensor {

  case class

hvac_cls(Date:String,Time:String,TargetTemp:Int,ActualTemp:Int,System:Int,SystemAge :Int,BuildingId:Int)


  case class

building(buildid:Int,buildmgr:String,buildAge:Int,hvacproduct:String,Country:String
)
```

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

    //Let us create a spark session object

    val spark = SparkSession

      .builder()

      .master("local")

      .appName("Spark SQL basic example")

      .config("spark.some.config.option", "some-value")

      .getOrCreate()

    println("Spark Session Object created")

    // Below statement will suppress all warnings
    spark.sparkContext.setLogLevel("WARN")
```

__Output :__

Spark Session Object created

Then we have loaded data from **HVAC** csv file and used count() function to calculate number of rows in the file and we have printed it.

Then we have taken first (header) row from this file and filtered out header row from it.

__Scala code :__

```
val data =
spark.sparkContext.textFile("C:\\AcadGild
Hadoop\\Assignments\\HVAC.csv");

println("HVAC Row Count->>"+data.count())

val header = data.first()

val data1 = data.filter(row => row != header)
```

```
println("Header removed from the data !")
```

Output :

HVAC Row Count->>8001

Header removed from the data !

Here we have converted **data1** RDD to DataFrame and then we have registered it as **HVAC** table.

Scala code :

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


val hvac = data1.map(x=>x.split(",")).map(x =>

hvac_cls(x(0),x(1),x(2).toInt,x(3).toInt,x(4).toInt,x(5).toInt,x(6).toInt)).toDF()


hvac.show()


println("HVAC Dataframe created !")


hvac.registerTempTable("HVAC")



println("Dataframe has been registered as HVAC table !")
```

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


HVAC Dataframe created !

Dataframe has been registered as HVAC table !
```

Here we have used **sql** transformation to create sql query from **HVAC** table and printed the result from this query. Then we have registered this hvac1 RDD as **HVAC1** table.

Scala code :

```
val hvac1 = spark.sql("select *,IF((targettemp - actualtemp) > 5, '1', IF((targettemp - actualtemp) <
- 5, '1', 0)) AS tempchange from HVAC")

hvac1.show()

hvac1.registerTempTable("HVAC1")

println("Data Frame has been registered as HVAC1 table !")
```

Output :

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

Data Frame has been registered as HVAC1 table !
```

Then we have loaded data from **building** csv file.

Then we have taken first (header) row from this file and filtered out header row from it. After this we have converted data3 RDD into build dataframe. Then we have printed this dataframe by using **show** function.

Scala code :

```scala
//Now lets load the second data set

val data2 = spark.sparkContext.textFile("C:\\AcadGild
Hadoop\\Assignments\\building.csv");

val header1 = data2.first()

val data3 = data2.filter(row => row != header1)

println("Header has been removed from the building data")

//Now let us create the building dataframe
```

```
val build = data3.map(x=> x.split(",")).map(x =>
building(x(0).toInt,x(1),x(2).toInt,x(3),x(4))).toD
F


build.show()
```

Output :

```
Header has been removed from the building data +----
---+--------+--------+-----------+------------+
|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|
+ -------+--------+-------- +---------- +------------ +
```

Here we have registered **building** data as buiding table.

Then we have used **sql** transformation to create sql query by making join of **HVAC1** with **building** table with help of buildingid and printed the result from this query.

Scala code :

```
build.registerTempTable("building")

println("Buildings data has been registered as building table")



//Now join the two tables

val build1 = spark.sql("select h.*, b.country, b.hvacproduct from building b
join hvac1 h on b.buildid = h.buildingid")

build1.show()
```

Output :

```
Buildings data has been registered as building table

+ -------+--------- +----------- +-----------+------+--------- +----------    +----------   + -------+----------        +

|     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
```

The we have selected only temperaturechange and country columns from build1 RDD and printed the result from this query.

Scala code :

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

val tempCountry = build1.map(x => (new Integer(x(7).toString),x(8).toString))

tempCountry.show()
```

Output :

```
+---+-------+
| _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
```

Then we have selected only those records for which temperaturechange = 1. i.e. we have selected only those records for which there is a difference in actual temperature and target temperature is greater than 5 and then we have printed the result.

Scala code :

```
//Filter the values

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

tempCountryOnes.show()
```

Output :

```
+---+-------+
| _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
```

Then we have grouped it by country and taken the count and then we have printed the result.

Scala code :

```
tempCountryOnes.groupBy("_2").count.show
```

Output :

[redacted]

```
+------------+   -----+
|        _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|
+------------+   -----+
```

Scala Program :

```
import org.apache.spark.sql.SparkSession

object Case_Study_3_Sensor {

  case class
hvac_cls(Date:String,Time:String,TargetTemp:Int,ActualTemp:Int,System:Int,SystemAge
:Int,BuildingId:Int)

  case class
building(buildid:Int,buildmgr:String,buildAge:Int,hvacproduct:String,Country:String
)



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

    //Let us create a spark session object

    val spark = SparkSession

      .builder()
      .master("local")
      .appName("Spark SQL basic example")
      .config("spark.some.config.option", "some-value")
      .getOrCreate()

    println("Spark Session Object created")

    // Below statement will suppress all warnings
    spark.sparkContext.setLogLevel("WARN")

    val data = spark.sparkContext.textFile("C:\\AcadGild
Hadoop\\Assignments\\HVAC.csv");

    println("HVAC Row Count->>"+data.count())

    val header = data.first()
    val data1 = data.filter(row => row != header)


    println("Header removed from the data !")




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


    val hvac = data1.map(x=>x.split(",")).map(x =>

hvac_cls(x(0),x(1),x(2).toInt,x(3).toInt,x(4).toInt,x(5).toInt,x(6).toInt)).toDF()


    hvac.show()


    println("HVAC Dataframe created !")


    hvac.registerTempTable("HVAC")



    println("Dataframe has been registered as HVAC table !")


    val hvac1 = spark.sql("select *,IF((targettemp - actualtemp) > 5, '1',
IF((targettemp - actualtemp) < -5, '1', 0)) AS tempchange from HVAC")


    hvac1.show()


    hvac1.registerTempTable("HVAC1")
```

```
println("Data Frame has been registered as HVAC1 table !")



    //Now lets load the second data set


    val data2 =
spark.sparkContext.textFile("C:\\AcadGild
Hadoop\\Assignments\\building.csv");


    val header1 = data2.first()


    val data3 = data2.filter(row => row != header1)



    println("Header has been removed from the building data")


    //Now let us create the building dataframe


    val build = data3.map(x=> x.split(",")).map(x
=>
building(x(0).toInt,x(1),x(2).toInt,x(3),x(4))).toD
F


    build.show()


    build.registerTempTable("building")


    println("Buildings data has been registered as building table")



    //Now join the two tables

    val build1 = spark.sql("select h.*, b.country, b.hvacproduct from building
b join hvac1 h on b.buildid = h.buildingid")


    build1.show()


    //Select temperature and country column from above


    val tempCountry = build1.map(x => (new Integer(x(7).toString),x(8).toString))
    tempCountry.show()

    //Filter the values

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

    tempCountryOnes.show()

    tempCountryOnes.groupBy("_2").count.show

  }

}
```

Complete Output :

```
Spark Session Object created
HVAC Row Count->>8001
Header removed from the data !

+----------+--------     +----------+----------+------+---------+----------  +
|      Date|   Time|TargetTemp|ActualTemp|System|SystemAge|BuildingId|
+----------+--------     +----------+----------+------+---------+----------  +
  |06-01-2013|00:00:01|       66|       58|  13|       20|       4|
  |06-02-2013|01:00:01|       69|       68|   3|       20|      17|
  |06-03-2013|02:00:01|       70|       73|  17|       20|      18|
  |06-04-2013|03:00:01|       67|       63|   2|       23|      15|
  |06-05-2013|04:00:01|       68|       74|  16|        9|       3|
  |06-06-2013|05:00:01|       67|       56|  13|       28|       4|
  |06-07-2013|06:00:01|       70|       58|  12|       24|       2|
  |06-08-2013|07:00:01|       70|       73|  20|       26|      16|
  |06-09-2013|08:00:01|       66|       69|  16|        9|       9|
  |06-10-2013|09:00:01|       65|       57|   6|       5|        12|
  |06-11-2013|10:00:01|       67|       70|  10|       17|      15|
  |06-12-2013|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

HVAC Dataframe created !
```

Dataframe has been registered as HVAC table !

```
+ ---------+--------      + ----------+----------+------+---------+----------+   ----------+
|       Date|   Time|TargetTemp|ActualTemp|System|SystemAge|BuildingId|tempchange|
+ ---------+--------      + ----------+----------+------+---------+----------+   ----------+
  |06-01-2013|00:00:01|       66|       58|  13|      20|       4|         1|
  |06-02-2013|01:00:01|       69|       68|   3|       20|      17|         0|
  |06-03-2013|02:00:01|       70|       73|  17|      20|      18|         0|
  |06-04-2013|03:00:01|       67|       63|   2|       23|      15|         0|
  |06-05-2013|04:00:01|       68|       74|  16|        9|       3|         1|
  |06-06-2013|05:00:01|       67|       56|  13|      28|       4|         1|
  |06-07-2013|06:00:01|       70|       58|  12|      24|       2|         1|
  |06-08-2013|07:00:01|       70|       73|  20|      26|      16|         0|
  |06-09-2013|08:00:01|       66|       69|  16|        9|       9|         0|
  |06-10-2013|09:00:01|       65|       57|   6|      5|        12|         1|
  |06-11-2013|10:00:01|       67|       70|  10|      17|      15|         0|
  |06-12-2013|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

Data Frame has been registered as HVAC1 table !
Header has been removed from the building data


+-------+--------      +-------- +-----------         +-----------        +
|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 has been registered as building table

```
+----------+--------          +----------    +---------- +------    +--------- +----------  +----------  +-------+-----------        +
|      Date|
Time|TargetTemp|ActualTemp|System|SystemAge|BuildingId|tempchange|country|hvacproduct|
+----------+--------          +----------    +---------- +------    +--------- +----------  +----------  +-------+-----------        +
|06-10-2013|09:00:01|        65|        57|     6|        5|        12|        1|Finland|       FN39TG|
|    6/18/13|23:13:19|      66|        75|      1|      13|          12|       1|Finland|       FN39TG|
|06-02-2013|13:43:51|        65|        72|    20|       26|          12|       1|Finland|       FN39TG|
|    6/13/13|00:13:20|      67|        77|      8|      19|          12|       1|Finland|       FN39TG|
|    6/16/13|03:13:20|       67|     55|     11|       16|      12|        1|Finland|     FN39TG|
|    6/30/13|17:13:20|       65|     57|     17|        9|      12|        1|Finland|     FN39TG|
|06-01-2013|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|
|06-05-2013|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|
|06-04-2013|21:13:20|       66|       72|    7|         1|      12|        1|Finland|     FN39TG|
|06-03-2013|02:00:01|       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|07:43:51|       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|
|06-10-2013|03:33:07|       70|       78|    5|         9|      12|        1|Finland|     FN39TG|
+ ----------+---------+----------+-----------+------ +---------+----------   +----------   + -------+----------        +
only showing top 20 rows


+---+-------+
| _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
```

```
+---+-------+

| _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
```

```
+------------+-----+

|          _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|
+------------+-----+
```