Code

Vivek Pandiyan

```
import org.apache.Spark.sql.functions.
import org.joda.time.format.DateTimeFormat.
Val inputPath = "C:/Users/vivek/Downloads/Pressure.csv"
Val Pressure = Sql Context.read.
                format ("com.databricks.Spark.csv")
                .option ("header", "true")
                .option ("delimiter", ",")
                .option ("inferSchema", "true")
                .load (inputpath)
Val temperature : Sql Context.read(
                .format ("com.databricks.Spark.csv")
                .option ("header", "true")
                .option ("delimiter", ",")
                .option ("inferschema", "true")
                .load (inputpath)
Val inputdew: "C:/Users/vivek/downloads/dew.csv"
Val dew = Sql.context.read.
                .format ("com.databricks.Spark.csv").
                .option ("header", "true")
                .option ("delimiter", ",")
                .option ("inferschema", "true")
                .load (inputdew)
Val inputhum = "C:/users/vivek/downloads/downdity.csv"
Val humidity : Sql.Context.read
                .format ("com.databricks.Spark.csv
                .option ("header", "true")
                .option ("delimiter", ",")
                .option ("inferschema", "true").
```

Viren Pandiyan

```
                    .load(inputpath)
val inputWinddd = "c:/Users/Viren/downloads/windDirection.csv"
val winddir = Sql.context.read.
                    .format("com.databricks.Spark.com")
                    .format("header", "true")
                    .format("delimiter", ",")
                    .Option("inferSchema", "true")
                    .load(inputpath).
                    .load(inputwinddd).

val inputwinds. = "c:/Users/Viren/downloads/windSpeed.csv"
val winds = Sql.context.read
                    .format("com.databricks.Spark.com")
                    .Option("header", "true")
                    .option("delimiter", ",")
                    .option("inferSchema", "true")
                    .load(inputwinds)
val inputVis = "C:/users/Viren/downloads/Visibility.csv"
val Visibility = Sql.context.read.
                    .format("com.databricks.Spark.com")
                    .Option("header", "true")
                    .option("delimiter", ",")
                    .Option("inferSchema", "true")
                    .load(inputVis)
```

Vivek Panchigar

```
import matplotlib
import matplotlib.pyplot as plt
import StringIO
import pandas as pd.
import Seaborn as sns
df = sqlContext.sql("Select Timestamp, Pressure from Pressure
                     group by timestamp").
%.pyspark.
heat map [0:100].
from pyspark.sql import SQLContext
df = sqlContext.table("Pressure").
df.head().
import os
import pandas as pd.
list = []
for row in glob.glob("c:/users/Vivek/downloads/pressure.csv")
    data = pd.read_csv(row)
data.count().
%pyspark.
def show(p):-
img = StringIO.StringIO()
P.savefig(img, format='svg')
img.seek(0)
```

(75) Print "%html" + img.buf. Viren Pandyan

(76). df = SqlContext.sql (" Select Pressure, timpstart,

(77) Month, count (*) as No of reading, Max (temperature),

(78) Min (temperature) from Pressure P inner join

(79) temperature T on P.timestamp = T.timestamp)

(80) Value = "Max temp"

(81) grouping = ["Month"]

(82) Val Fireday = List ("01/01/2016", "01/16/2016",

(83) "01/20/2016", "02/21/2016",)

(84) "03/07/2016, 03/07/2016)

(85) def Forest_Fire : (String Array [Double]) = {

(86) Vals values = Array (Day . to double, humidity. to double

(87) winds. to double, Temp . to double

(88) days from - max temp (year. to int,

(89) month. to int, day. to int))

(90) - import java.io. _

(91) import org. joda. time. _

(92) import org. joda. time. format. _

(93) import org. joda. time frame . Date time Format

(94) import . org. joda. time. Date time

(95) import org. joda. time . Days.

Vivek Pandiyan

```
def get-date ( year: Int , month: Int , day: Int ) =
  "%04d%02d%02d".format(year, month, day)
%sql
Select  timestamp, temperature, Pressure, humidity
        wind speed , wind direction, Visibility from
        Forest fire  ~~groupby Ita~~
        order by timestamp.
%sql
Select  max(temperature), Month(timestamp) from
        temperature group by Month(Timestamp).
        Order by month (timestamp).
%sql.
Select. date (T.timestamp), day (F.timpstamp),
        Max (F.temperature), Max(F.pressure),
        From temperature T join Forest Fire F
        day (F.Timpestamp)=day(T.timestamp.)
%pyspark.
from pandas import Series , Dataframe.
import pandas as pd.
import numpy as np.
dataframe ( pressure ).
```