Lab 7

In this lab, we analyze historical data about the stations of the bike sharing system of Barcelona. Your task consists in identifying the most "critical" timeslot (day of the week, hour) for each station and store the result in a KML file that can be visualized on a map.

The analysis is based on two files (available in the shared folder of the BigData@Polito cluster):

1. /share/students/bigdata/Dati/Lab7/datiCompleti/register.csv
2. /share/students/bigdata/Dati/Lab7/datiCompleti/stations.csv

1) register.csv contains the historical information about the number of used and free slots for ~3000 stations from May 2008 to September 2008. Each line of register.csv corresponds to one reading about the situation of one station at a specific timestamp. Each line has the following format:

   *stationId***\t***timestamp***\t***usedslots***\t***freeslots*

   where *timestamp* has the format *date time*

   For example, the line

   23 2008-05-15 19:01:00 5 13

   means that there were **5** used slots and **13** free slots at station **23** on **May 15, 2008** at **19:01:00**.

   Pay attention that the first line of register.csv contains the header of the file. Moreover, **some of the lines of register.csv contain wrong data** due to temporary problems of the monitoring system. Specifically, some lines are characterized by used slots = 0 and free slots = 0. Those lines must be filtered before performing the analysis.

2) stations.csv contains the description of the stations. Each line of registers.csv has the following format:

   *stationId***\t***longitude***\t***latitude***\t***name*

   For example, the line

   1 2.180019 41.397978 Gran Via Corts Catalanes

   contains the information about station **1**. The coordinates of station 1 are 2.180019,41.397978 and its name is **Gran Via Corts Catalanes**.

Task 1

Write a single Spark application that identifies the most "critical" timeslot for each station. This analysis can support the planning of the rebalancing operations among stations.

In this application, each combination "day of the week – hour" is a timeslot and is associated with all the readings associated with that combination, independently of the date. For instance, the timeslot “Wednesday - 15” corresponds to all the readings made on Wednesday from 15:00:00 to 15:59:59, independently of the date.

A station Si is in the critical state in a specific timestamp if the number of free slots is equal to 0 (i.e., the station is full).

The “criticality” of a station Si in the timeslot Tj is defined as

$$\frac{number\ of\ readings\ with\ num.of\ free\ slot\ equal\ to\ 0\ for\ the\ pair\ (Si,Tj)}{total\ number\ of\ readings\ for\ the\ pair\ (Si,Tj)}$$

Write an application that:

- Computes the criticality value for each pair (Si,Tj).
- Selects only the pairs having a criticality value greater than or equal to a minimum criticality threshold. The minimum criticality threshold is an argument of the application.
- Selects the most critical timeslot for each station (consider only timeslots with a criticality greater than or equal to the minimum criticality threshold). If there are two or more timeslots characterized by the highest criticality value for the same station, select only one of those timeslots. Specifically, select the one associated with the earliest hour. If also the hour is the same, consider the lexicographical order of the name of the week day.
- Stores in one single (KML) file the information about the most critical timeslot for each station. Specifically, the output (KML) file must contain one marker of type Placemark for each pair (Si, most critical timeslot for Si) characterized by the following features:
    - StationId
    - Day of the week and hour of the critical timeslot
    - Criticality value
    - Coordinates of the station (longitude, latitude)

  Do not include in the output (KML) file the stations for which there are no timeslots satisfying the minimum criticality threshold.

Hints

- To extract hour and weekday from a timestamp, consider using the datetime package of Python (e.g., functions strptime() and strftime())
    - Example

      ```
      from datetime import datetime
      timestamp = "2008-05-15 12:01:00"
      datetimeObject = datetime.strptime(timestamp, "%Y-%m-%d %H:%M:%S")
      dayOfTheWeek = datetimeObject.strftime("%a")
      hour = datetimeObject.hour
      ```
- To create one single output file, set the number of partitions of the final RDD to 1 by using coalesce(1) before invoking the saveAsTextFile() method.

The output (KML) file must have the following format (one KML Placemark per line):

```
<Placemark><name>44</name><ExtendedData><Data
name="DayWeek"><value>Mon</value></Data><Data
name="Hour"><value>3</value></Data><Data
name="Criticality"><value>0.5440729483282675</value></Data></ExtendedData><
Point><coordinates>2.189700,41.379047</coordinates></Point></Placemark>
<Placemark><name>9</name><ExtendedData><Data
name="DayWeek"><value>Sat</value></Data><Data
name="Hour"><value>10</value></Data><Data
name="Criticality"><value>0.5215827338129496</value></Data></ExtendedData><
Point><coordinates>2.185294,41.385006</coordinates></Point></Placemark>
```

Copy and paste the output inside a KML file formatted as follows:

```
<kml xmlns="http://www.opengis.net/kml/2.2"><Document>
        Copy and paste here the output generated by your application
</Document></kml>
```

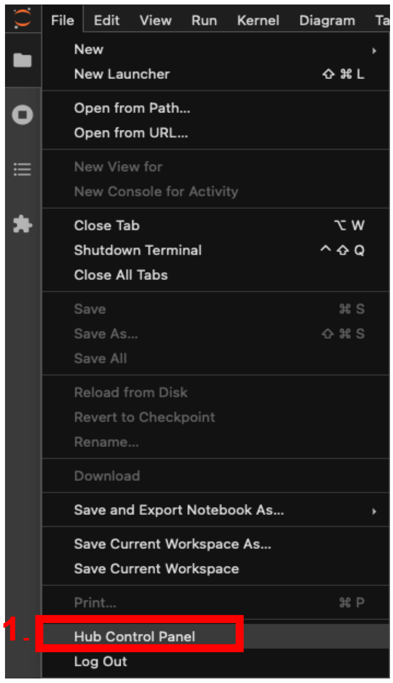
The content of the KML file can be used to show on a Google map the stations characterized by a high criticality value and the associated information. You can visualize the result of your analysis online on a map by uploading your KML file on one of the following web pages:

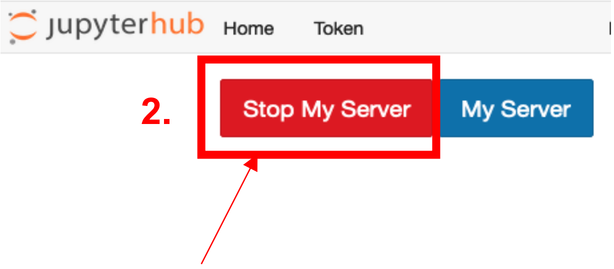
- https://kmlviewer.nsspot.net
- https://www.gpsvisualizer.com

⚠⚠⚠ Shut down JupyterHub container ⚠⚠⚠

As soon as you complete all the tasks and activities on JupyterHub environment, please remember to shut down the container to let all your colleagues in all the sessions connect on JupyterHub and do all the lab activities.

1. Go into File -> Hub Control Panel menu
2. A new browser tab opens with the “Stop My Server” button. Click on it and wait till it disappears.





Click the “Stop My Server” button