Assignment 3

Welcome to Assignment 3. This will be even more fun. Now we will calculate statistical measures on the test data you have created.

YOU ARE NOT ALLOWED TO USE ANY OTHER 3RD PARTY LIBRARIES LIKE PANDAS. PLEASE ONLY MODIFY CONTENT INSIDE THE FUNCTION SKELETONS Please read why: https://www.coursera.org/learn/exploring-visualizing-iot-data/discussions/weeks/3/threads/skjCbNgeEeapeQ5W6suLkA (https://www.coursera.org/learn/exploring-visualizing-iot-data/discussions/weeks/3/threads/skjCbNgeEeapeQ5W6suLkA) . Just make sure you hit the play button on each cell from top to down. There are seven functions you have to implement. Please also make sure than on each change on a function you hit the play button again on the corresponding cell to make it available to the rest of this notebook. Please also make sure to only implement the function bodies and DON'T add any additional code outside functions since this might confuse the autograder.

So the function below is used to make it easy for you to create a data frame from a cloudant data frame using the so called "DataSource" which is some sort of a plugin which allows ApacheSpark to use different data sources.

All functions can be implemented using DataFrames, ApacheSparkSQL or RDDs. We are only interested in the result. You are given the reference to the data frame in the "df" parameter and in case you want to use SQL just use the "spark" parameter which is a reference to the global SparkSession object. Finally if you want to use RDDs just use "df.rdd" for obtaining a reference to the underlying RDD object.

Let's start with the first function. Please calculate the minimal temperature for the test data set you have created. We've provided a little skeleton for you in case you want to use SQL. You can use this skeleton for all subsequent functions. Everything can be implemented using SQL only if you like.

In [73]:
```
def minTemperature(df,spark):
    return spark.sql("SELECT MIN(temperature) as mintemp from washing").fir
```

Please now do the same for the mean of the temperature

In [74]:
```
def meanTemperature(df,spark):
    return spark.sql("SELECT AVG(temperature) as meantemp from washing").fi
```

Please now do the same for the maximum of the temperature

In [75]:
```
def maxTemperature(df,spark):
    return spark.sql("SELECT MAX(temperature) as maxtemp from washing").fir
```

Please now do the same for the standard deviation of the temperature

In [76]:
```
def sdTemperature(df,spark):
    return spark.sql("SELECT STDDEV(temperature) AS sdtemp FROM washing").f
```