By the end of this activity, you will be able to perform the following in Spark:

1. Remove rows containing missing values from a DataFrame.
2. Impute missing values with the average value.

In this activity, you will be programming in a Jupyter Python Notebook. If you have not already started the Jupyter Notebook server, see the instructions in the Reading *Instructions for Starting Jupyter*.

Step 1. **Open Jupyter Python Notebook.** Open a web browser by clicking on the web browser icon at the top of the toolbar:

https://d3c33hcgiwev3.cloudfront.net/imageAssetProxy.v1/RCneZE7PEeaqTxIkdCEfsw_c491f272226b35805e44abef7a7a22a9_browser-icon.png?expiry=1576108800000&hmac=LxE5WF1f27_DiO3b09IplBs1c9rWUs165BjzNgIfBkc

Navigate to *localhost:8889/tree/Downloads/big-data-4*:

https://d3c33hcgiwev3.cloudfront.net/imageAssetProxy.v1/9Zu58oqhEeaKKwpaECzIKQ_361b99533aaa8d7cde3e3df56b69b3f5_browser.png?expiry=1576108800000&hmac=Xf-EnpczSZaEP1-e1H_6AbBfXD-xaftjc9QB8CFSsqc

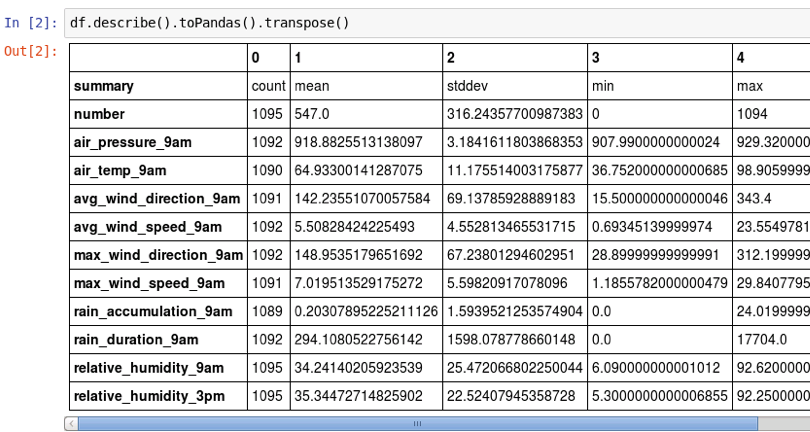
Open the handling missing values notebook by clicking on *handling-missing-values.ipynb:*

https://d3c33hcgiwev3.cloudfront.net/imageAssetProxy.v1/bhsBl4qhEeaoHhIiFgcrVw_8c9c7656c00cd1dab1f8ee8d410ee456_notebook.png?expiry=1576108800000&hmac=RC-VfJ4DV7p_dihxlckkCb-Pix_u8n5vBC2m_58an00

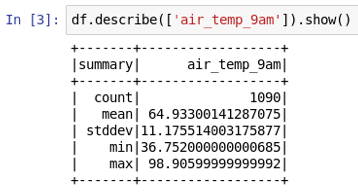
Step 2. **Load classes and weather data.** Run the first cell in the notebook to load the *SQLContext* class, create an instance of *SQLContext*, and read the weather data into a DataFrame.



Step 3. **Print summary statistics.** We can print the summary statistics for all the columns using *describe():*



Let's just look at the statistics for the air temperature at 9am:



This says that there are 1090 rows. The total number of rows in the DataFrame is 1095:

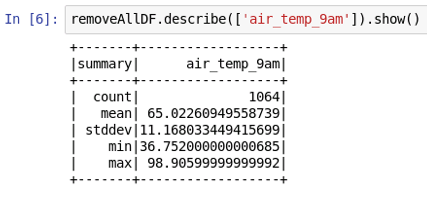


This means that 5 of the rows in the *air\_temp\_9am* column are missing values.

Step 4. **Remove missing values.** We can drop all the rows missing a value in any calling using *na.drop():*

https://d3c33hcgiwev3.cloudfront.net/imageAssetProxy.v1/S0_MT4qgEeaTPwpQDCu1cw_3ee72b641e98d4c96f24267503dddab5_remove-all.png?expiry=1576108800000&hmac=RQGJMaricRo1UuqyDObR5hf1WmTRO9YhP351Xpwb3gA

Let's look at the summary statistics for *air\_temp\_9am* with the missing values dropped:

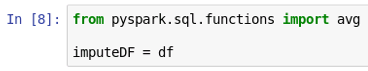


We can see that the mean and standard deviation is close to the original values: mean is 64.933 vs. 65.022, and standard deviation is 11.175 vs. 11.168.

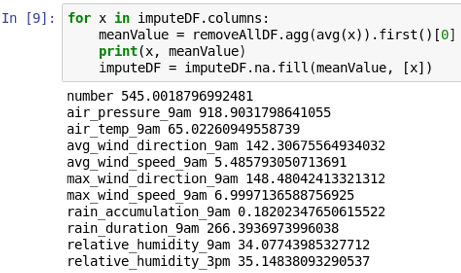
The count is 1064, which means that 1095 - 1064 = 31 rows were dropped. We can see this agrees with the total number of rows in the new DataFrame:



Step 5. **Impute missing values.** Instead of removing rows containing missing values, let's replace the values with the mean value for that column. First, we'll load the *avg* function and make a copy of the original DataFrame:



Next, we'll iterate through each column in the DataFrame: compute the mean value for that column and then replace any missing values in that column with the mean.

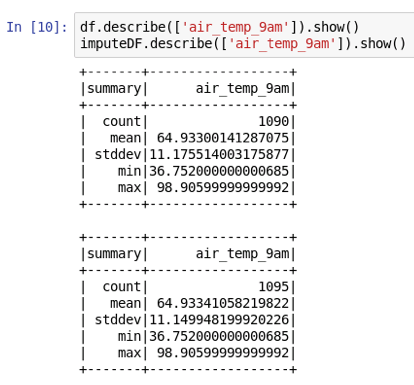


The *agg()* function performs an aggregate calculation on the DataFrame and *avg(x)* specifies to compute the mean on column *x*. The *agg()* function returns a DataFrame, *first()* returns the first *Row*, and *[0]* gets the first value.

The last line of code uses *na.fill()* to replace the missing values with the mean value (first argument) in column *x* (second argument).

The output of executing this cell prints the mean values for each column and we can see the mean value for *air\_temp\_9am* is the same as the mean when we removed all the missing values in step 4, *i.e.,* 65.022.

Step 6. **Print imputed data summary statistics.** Let's call *describe()* to show the summary statistics for the original and imputed *air\_temp\_9am:*



The count for the imputed data is larger since the 5 rows with missing data have replaced with real values. Additionally, we can see that the means are close, but not equal, and this is probably due to round-off error.