Explore 🗸

Data Models vs. Data Formats

Working with Streaming Data

Hands-On: Handling Data Streams

- Reading: Exploring
 Streaming Sensor Data
 10 min
- Video: Exploring Streaming Sensor Data
- Reading: Instructions for Creating a Twitter App (Optional) 10 min
- Reading: Exploring
 Streaming Twitter Data
 (Optional)
 10 min
- Video: Exploring Streaming Twitter Data (Optional) 4 min

By the end of this activity, you will be able to:

- 1. View semi-structured data streaming in real-time from a weather station
- 2. Create plots of streaming weather station data

Step 1. Open a terminal shell. Open a terminal shell by clicking on the square black box on the top left of the screen.



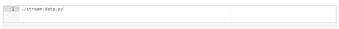
Change into the sensor directory:



Run Is to see the sensor data files and scripts:

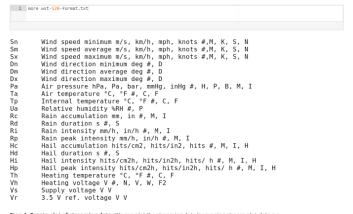
```
[cloudera@quickstart sensor]$ ls plot-data.py stream-plot-data.py wxt-520-format.txt stream-data.py wx-data.txt
```

Step 2. View streaming weather station data. Run stream-data.py to see streaming data from the weather station:

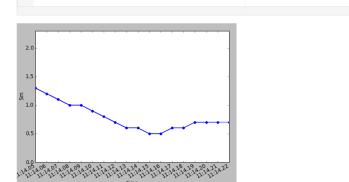


The measurements are appearing as they are produced by the weather station. By looking at the timestamp, we can see that they arrive about every second. Additionally, different measurement types are produced at different frequencies. For example R1 is measured every second, but R2 is less frequent.

 $\textbf{Step 3. View key for measurements.} \ \text{Run more wxt-} 520\text{-} for mat. txt to see the key for the measurement fields the search of the measurement fields the search of the measurement fields the search of the search of the search of the measurement fields the search of the sea$



Step 4. Create plot of streaming data. We can plot the streaming data by running stream-plot-data.py:



This show a plot of the average wind speed (Sm), and it updated every time a new measurement is produced. Notice that the plot is updated every second. We can create plots for other measurements by specifying the measurement in the last argument to stream-plot-data,py. For example, we can plot air temperature (Ta) by running:

```
1 ./stream-plot-data.py Tai
```

This plot is updated less frequently than the first plot since air temperature measurements are produced less frequently.

Mark as completed

6 P P