MONASH University

MONASH
INFORMATION
TECHNOLOGY

FIT5202 – Data Processing for Big Data

Granularity Reduction and Sensor Arrays

Edited by
Chee-Ming Ting (21 May 2021)

Developed by
Prajwol Sangat

GROUP OF EIGHT AUSTRALIA

Last Week

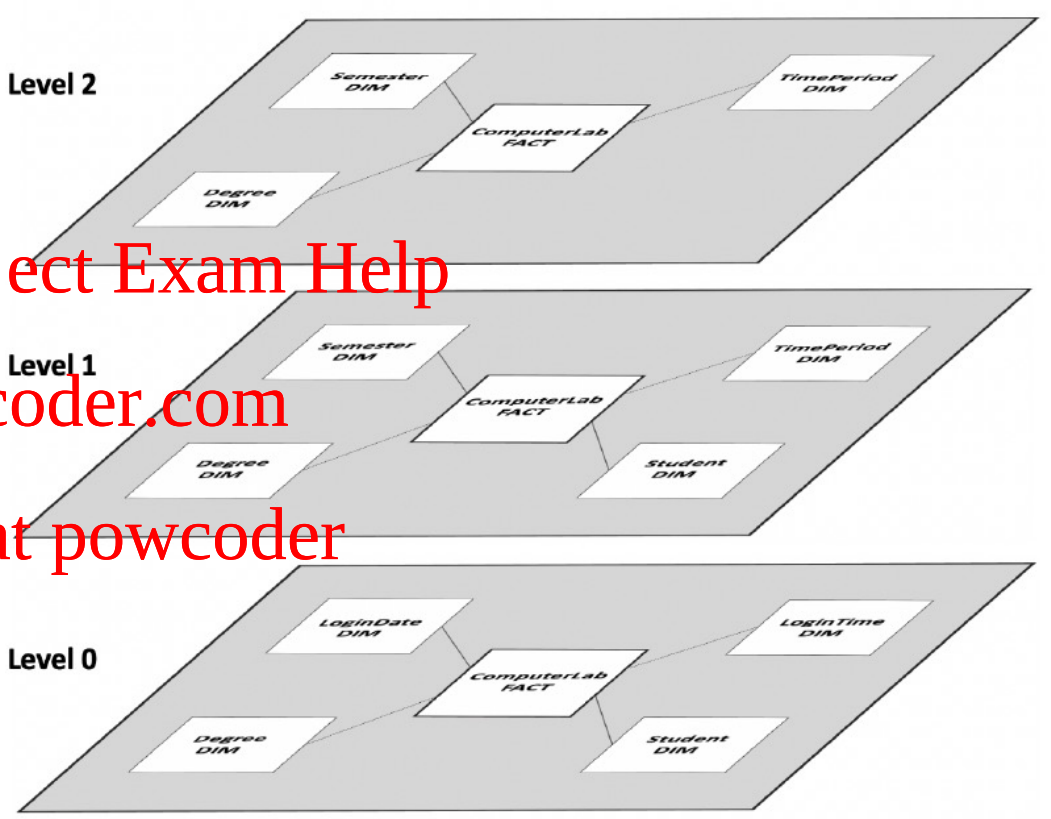
- Overview of Stream join
- Time based window stream join (Unbounded)
- Tuple based window stream join (Unbounded)
- Bounded stream join

This Week

- Granularity Reduction in Data Streams
- Mixed Levels of Granularity
- Sensor Arrays

Granularity

- **Granularity** is the level of detail at which **data** are stored in a database.
- level-0, the bottom level indicating no aggregation (raw data) → high granularity.
- level-1 and level-2 with more aggregation → lower granularity.
- Motivation of having multiple levels: Direct querying on higher level of aggregation is more efficient, considering the high volume & high velocity of data stream



Granularity Reduction in Data Streams

- Granularity is not only for efficient retrieval.
- Granularity is also about managing complexity.
  - higher granularity → raw data, complete but complex
  - lower granularity → data simplified (or summarized), less complex & less details

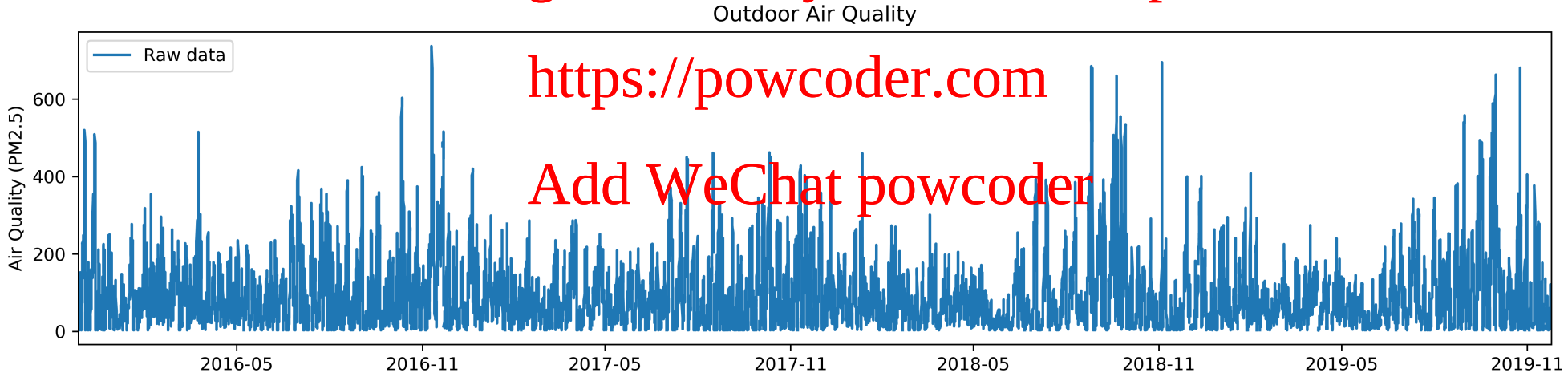


Figure1: Outdoor Air Quality of a city, measured in PM2.5 (raw data collected over hours)

Granularity Reduction in Data Streams

- Granularity is not only for efficient retrieval.
- Granularity is also about managing complexity.

To reduce granularity: Use mean function to aggregate hourly data into one day data

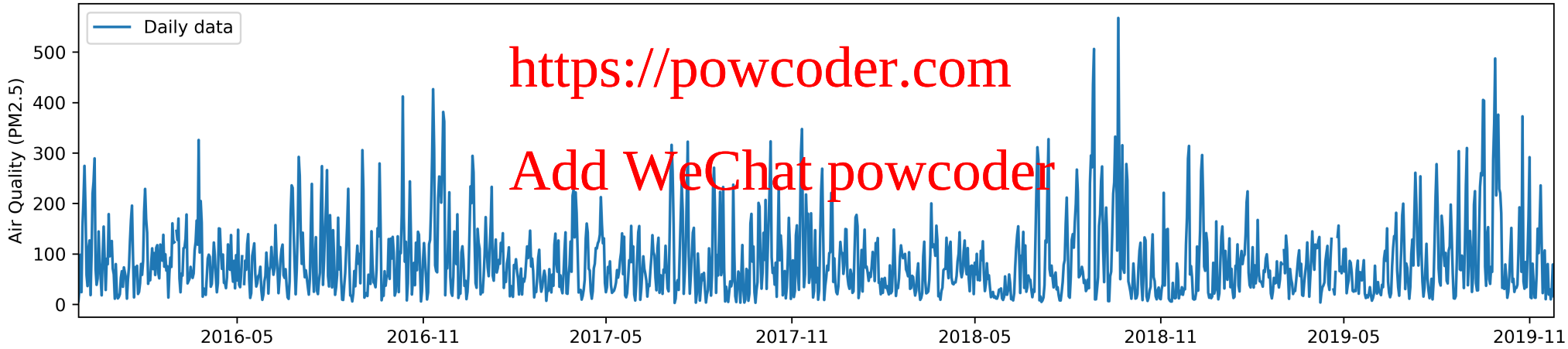


Figure2: Outdoor Air Quality of a city, measured in PM2.5 (Daily Aggregation)

Granularity Reduction in Data Streams

- Granularity is not only for efficient retrieval.
- Granularity is also about managing complexity.

Weekly aggregation – some trend is observed

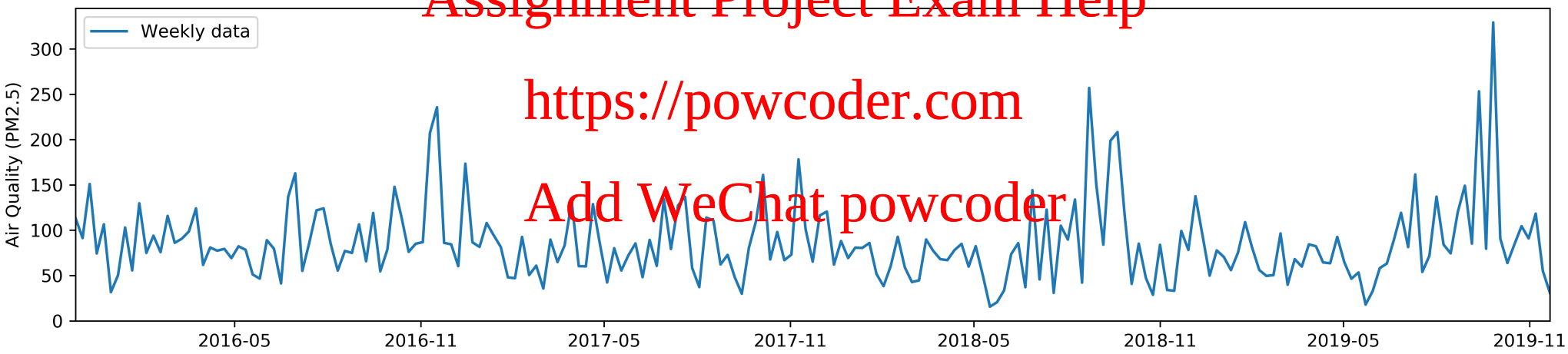


Figure3: Outdoor Air Quality of a city, measured in PM2.5 (Weekly Aggregation)

Granularity Reduction in Data Streams

- Granularity is not only for efficient retrieval.
- Granularity is also about managing complexity.
- Granularity may simplify the complexity of the information.

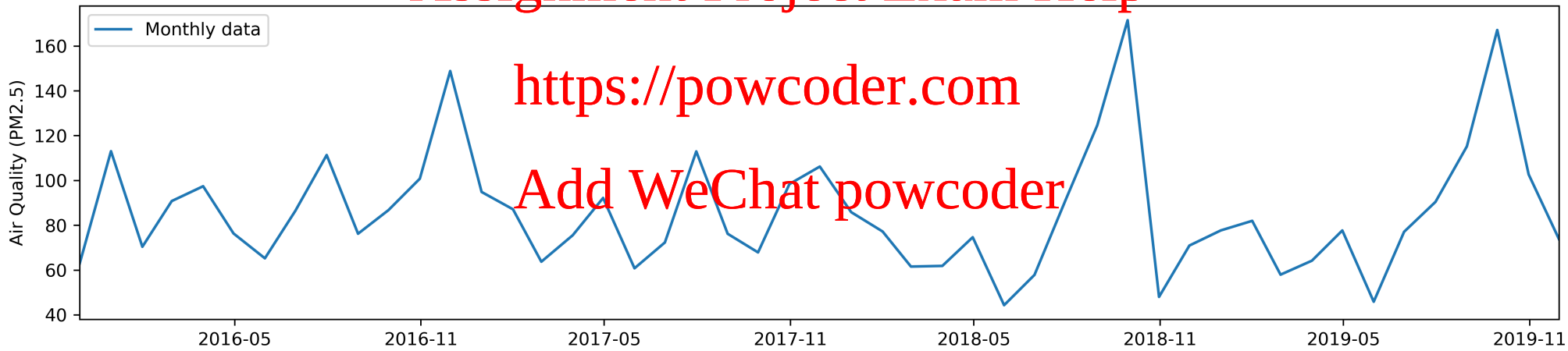


Figure4: Outdoor Air Quality of a city, measured in PM2.5 (Monthly Aggregation)

Trend of Covid cases in Malaysia

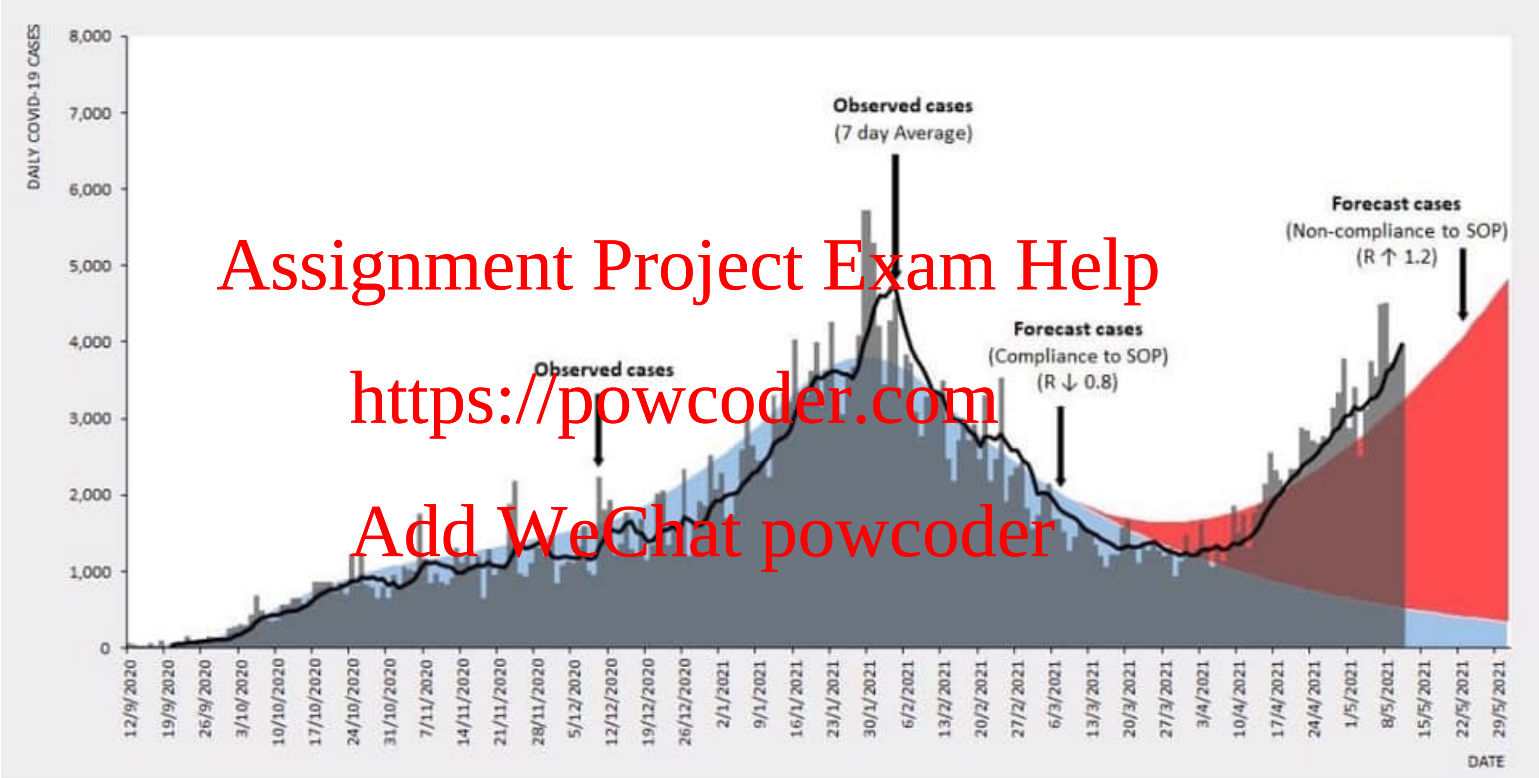


Image source: Ministry of health Malaysia

Fixed-Size Windows

- Time-based windowing is used to reduce granularity in data streams
  - Window size is uniform throughout the data stream.
  - Time-slide is applied at regular interval.

- Two time-slide approaches for fixed-size time-based windows:

- Overlapped Windows

  Slide time is less than the window size
- Non-overlapped Windows

  Slide time is equivalent to the window size

Fixed-Size Windows

- Overlapped Windows: No Granularity Reduction
  - When windowing is applied, normally, the aim is to calculate the mean (or any other statistical function) of values within the window.
  - For time-slide of 1 time unit/1 record → No reduction in terms of number of records. This is a pure moving average (also know as rolling mean).

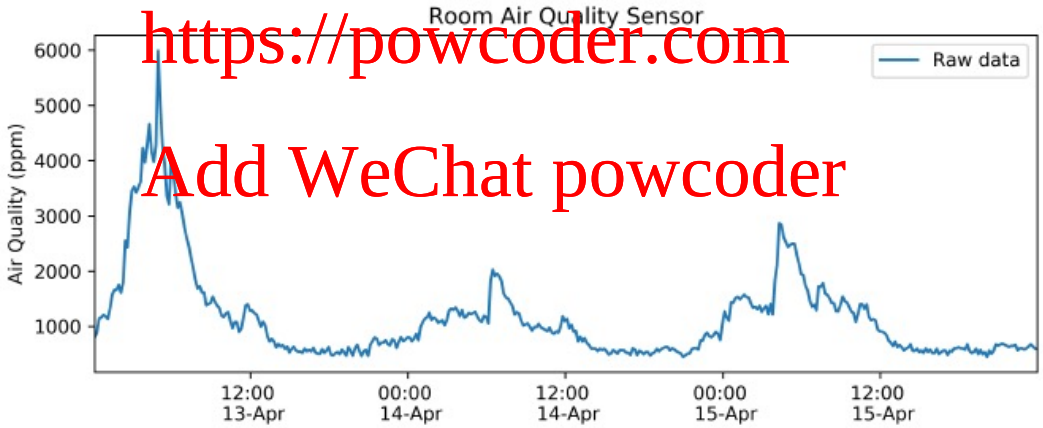
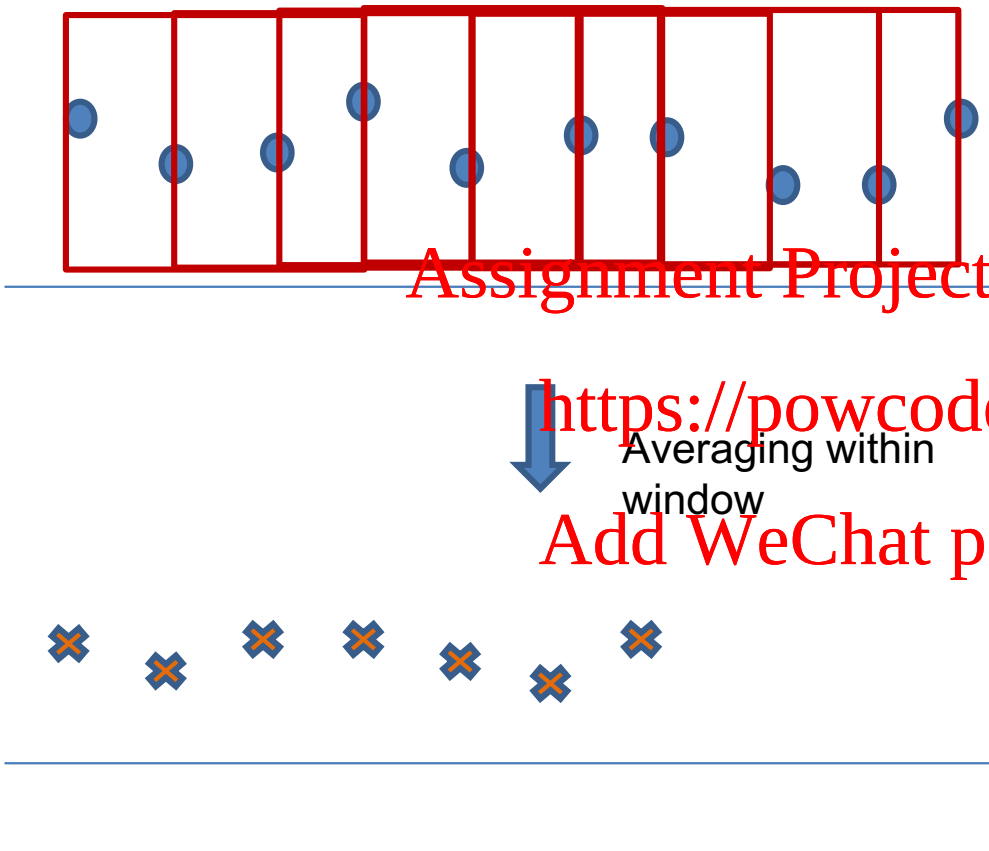


Figure5: The Indoor Air Quality for 3 days (10 mins sensor recording)

Overlapped Windows: No Granularity Reduction



Window size = 4 unit of time/ 4 record
Slide = 1 unit time / 1 record

Fixed-Size Windows

- Overlapped Windows: No Granularity Reduction

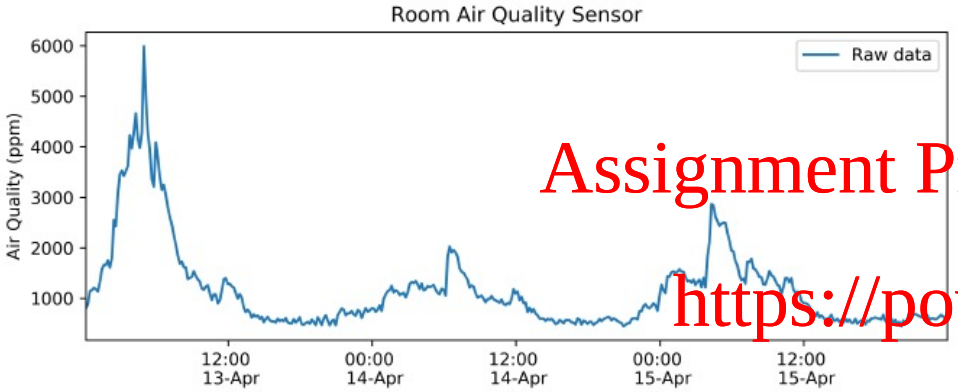


Figure5: The Indoor Air Quality for 3 days (10 mins sensor recording)

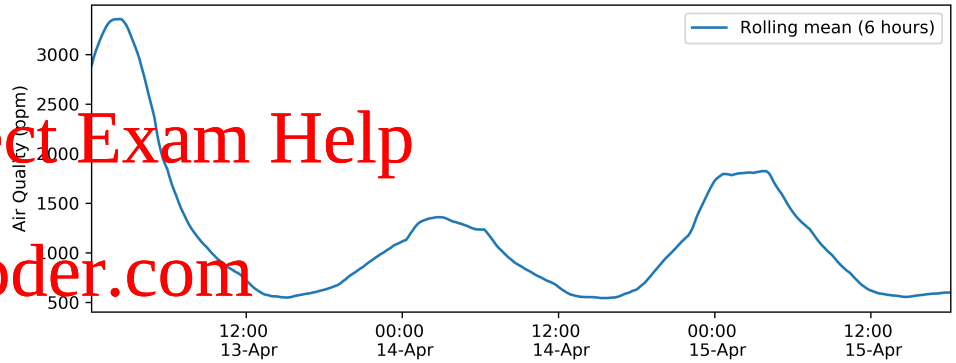


Figure6: Rolling mean of 6 hours (window size = 6 hours, time-slide every record/ 10 mins).

- Number of data points is not reduced
- Rolling mean (moving average) smooth out data points – revealing smoother trend
- The longer the window size – the smoother the trend

Fixed-Size Windows

- Overlapped Windows: With Granularity Reduction
  - When the time slide is more than one unit of time, there will be a reduction in terms of number of records after aggregation.

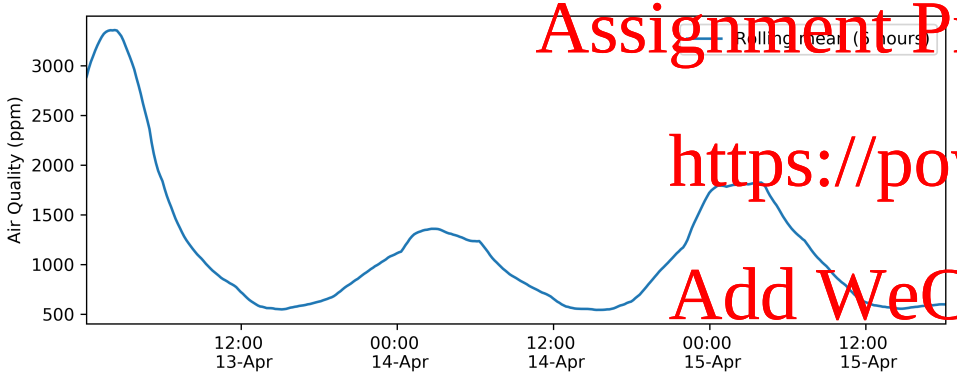


Figure6: Rolling mean of 6 hours.
(window size = 6 hours, time-slide =10 mins)

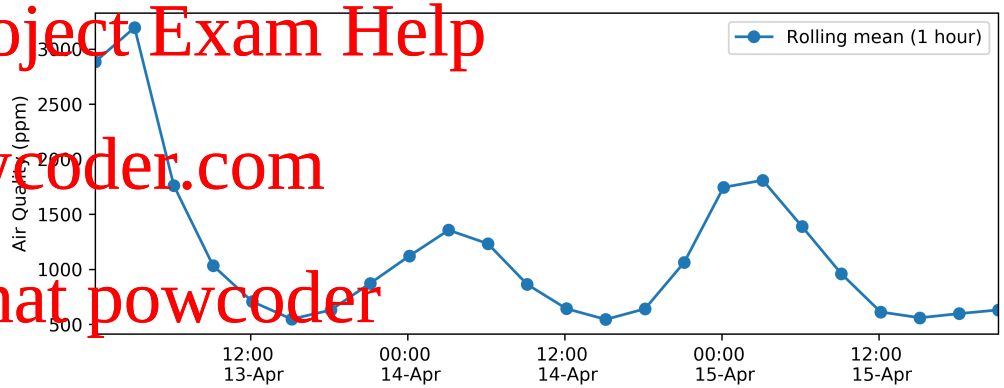
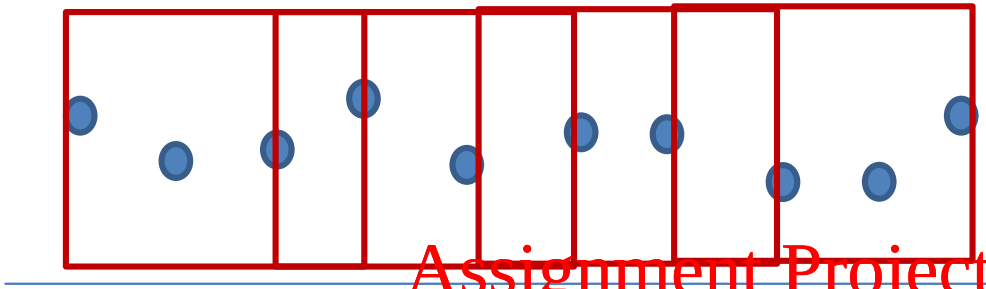


Figure7: Window size = 6 hours, time-slide = 3 hours.

→ Data will be produced every 3 hours, instead of every 10 mins

Overlapped Windows: Granularity Reduction



Window size = 4 unit of time/ 4 record
Slide = 2 unit time / 2 record

Time unit

Averaging within window

Time unit

Fixed-Size Windows

- Non-Overlapped Windows: Granularity Reduction
  - Consecutive windows are not overlapped, but there is no gap between the windows.

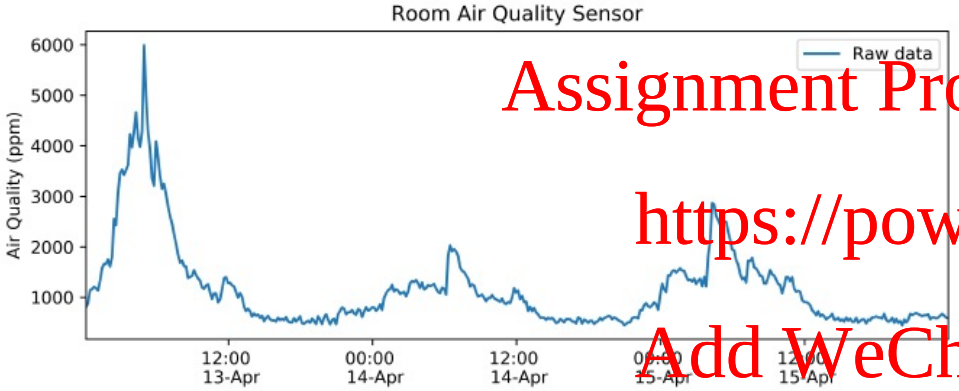


Figure5: The Indoor Air Quality for 3 days

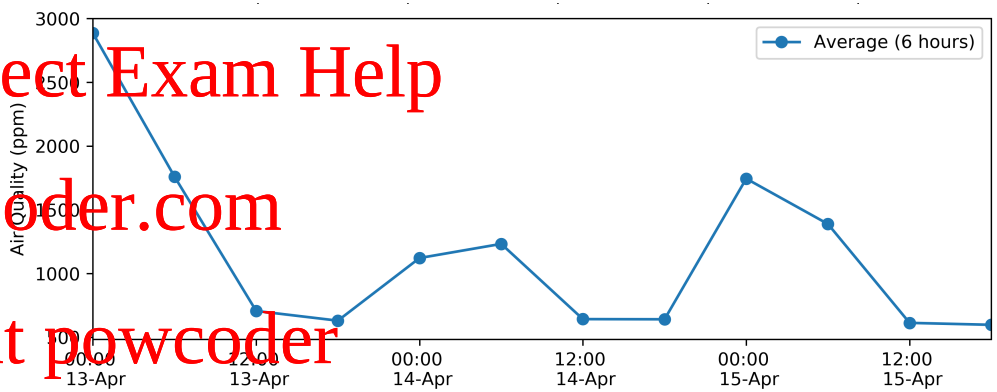
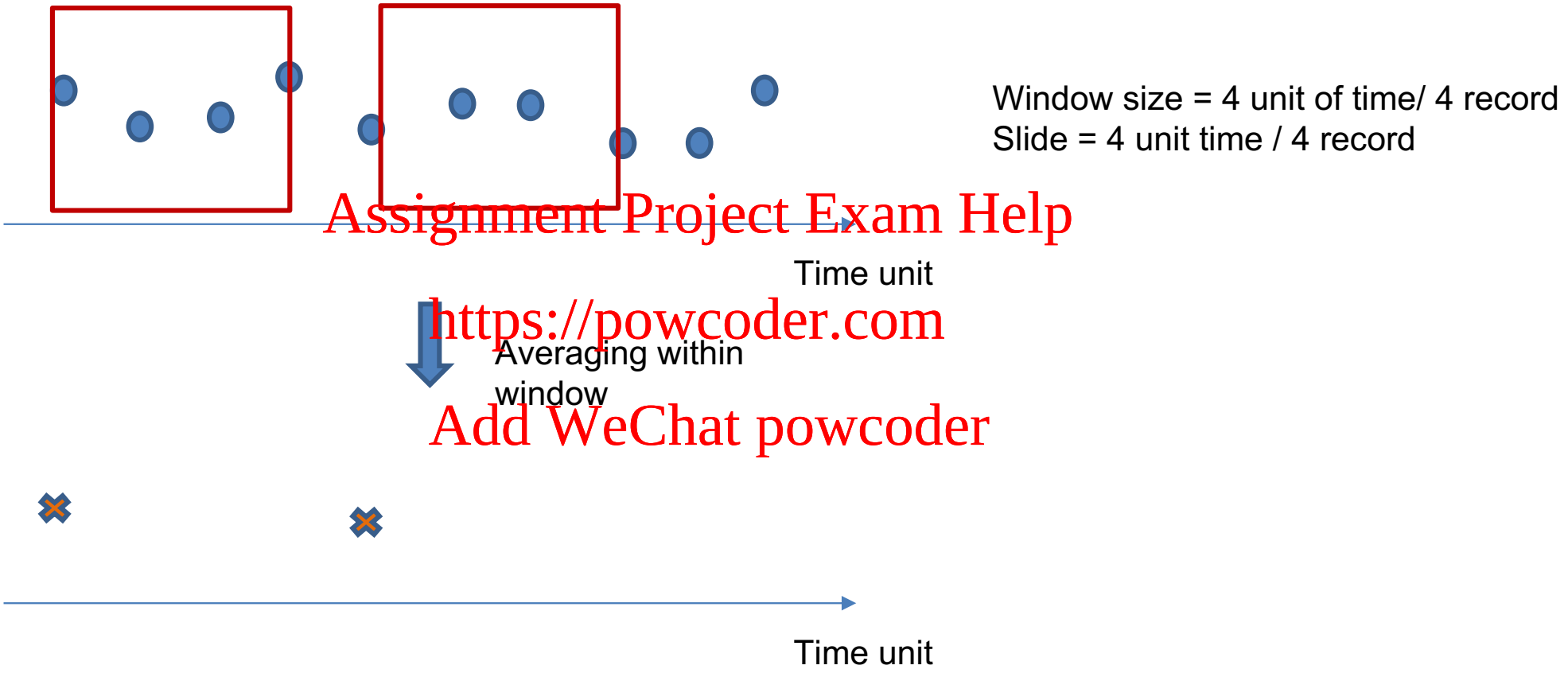


Figure8: The window size of 6 hours (time slide = 6 hours), and consecutive windows are not overlapped.

Besides averaging, the reduction function in the window can be any aggregate reduction function, such as average, min or max.

Non-Overlapped Windows: Granularity Reduction



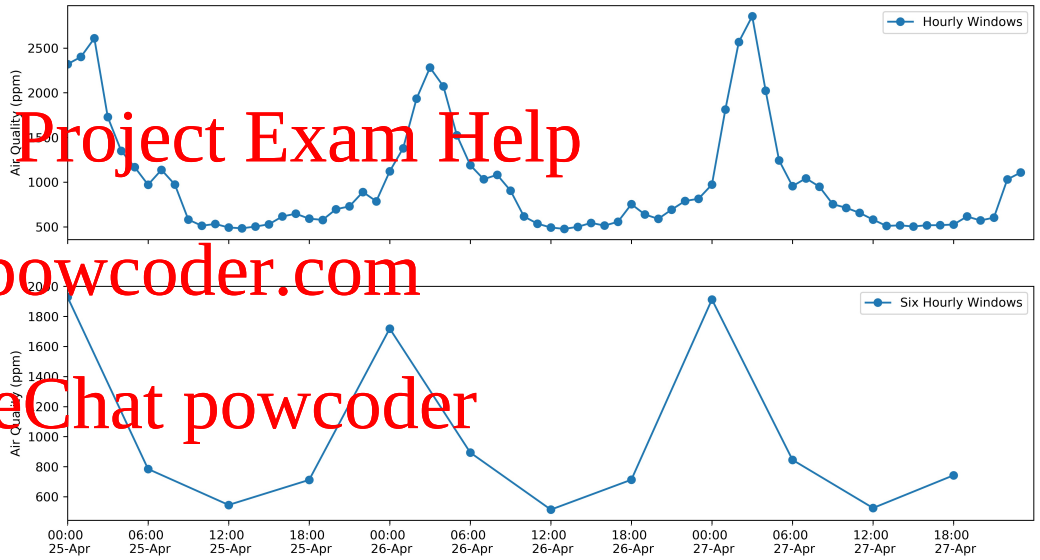
Mixed Levels of Granularity

- Different levels of granularity combined into one level.
- Mixed level of granularity can be two types:
  - Temporal-based Mixed Levels of Granularity
    - Time based.
  - Spatial-based Mixed Levels of Granularity
    - Space or location based.

Mixed Levels of Granularity

- Temporal-based Mixed Levels of Granularity
  - Time based.

A typical granularity reduction is shown in Figure 9 (top) and 10 (bottom), which are hourly granularity and 6-hourly granularity. These use fixed size windows.



1-hour window granularity:
- win size = 1 hour, slide = 1 hour

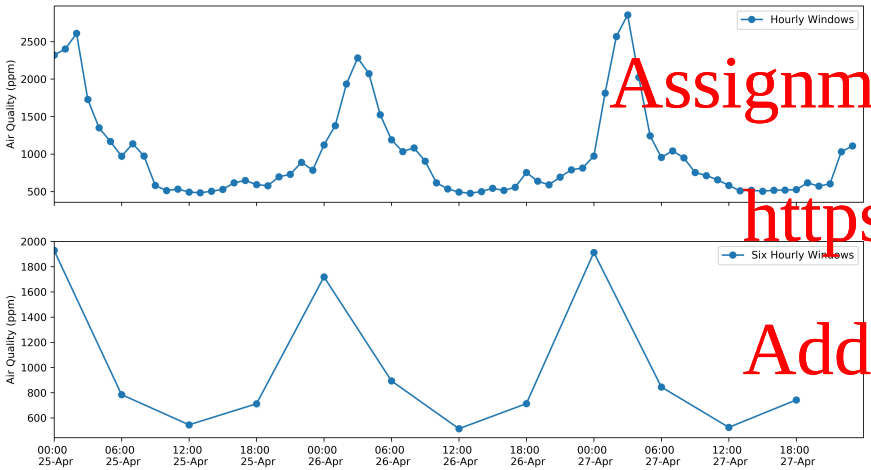
6-hour window granularity:
- win size = 6 hours, slide = 6 hours

Reveal seasonal trend

- hourly granularity has more details than the 6-hourly granularity

Mixed Levels of Granularity

- Temporal-based Mixed Levels of Granularity
  - Time based.



Mixing

Ex:
6am to midnight → use the 6-hourly granularity,
midnight to 6am → use the hourly granularity.

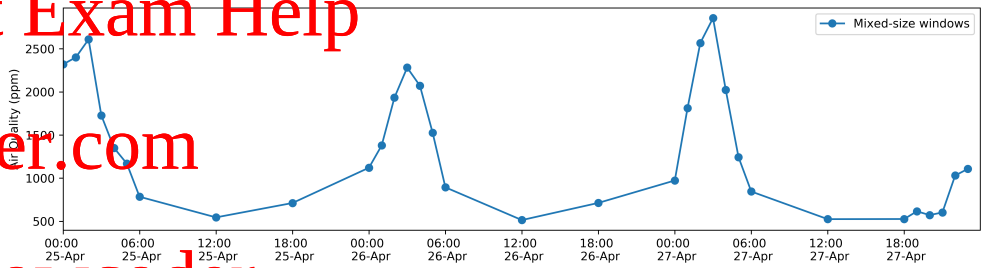


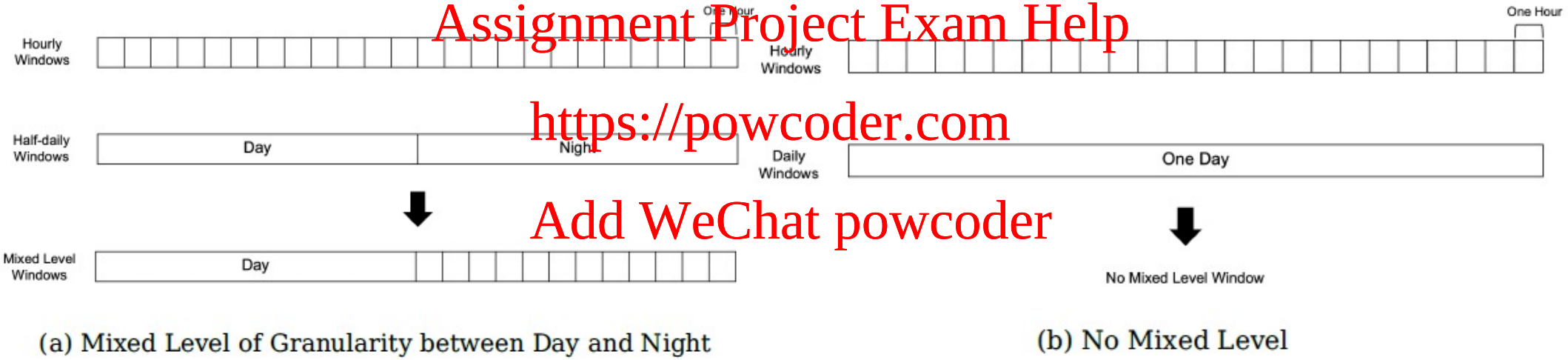
Figure11: mixing hourly granularity and 6-hourly grar
These use fixed size windows.

- ❑ Combine different levels of granularity
- ❑ It captures generality of 6 hour windows
- ❑ At certain period, granularity is further broken down into lower level

"Drill-down"
- Allow users to move from summary/aggregate information to detailed data

Mixed Levels of Granularity

- Temporal-based Mixed Levels of Granularity
  - Time based.



(a) Mixed Level of Granularity between Day and Night

(b) No Mixed Level

Require categorization of data

Mixed Levels of Granularity

- Spatial-based Mixed Levels of Granularity
  - Location based.

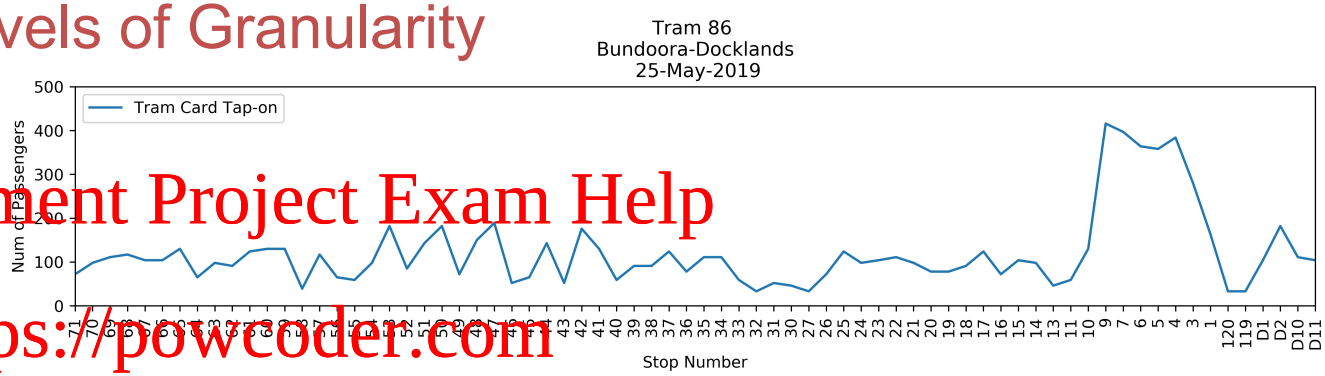


Figure 12: The tram data based on Stop Number

- Level-0 granularity

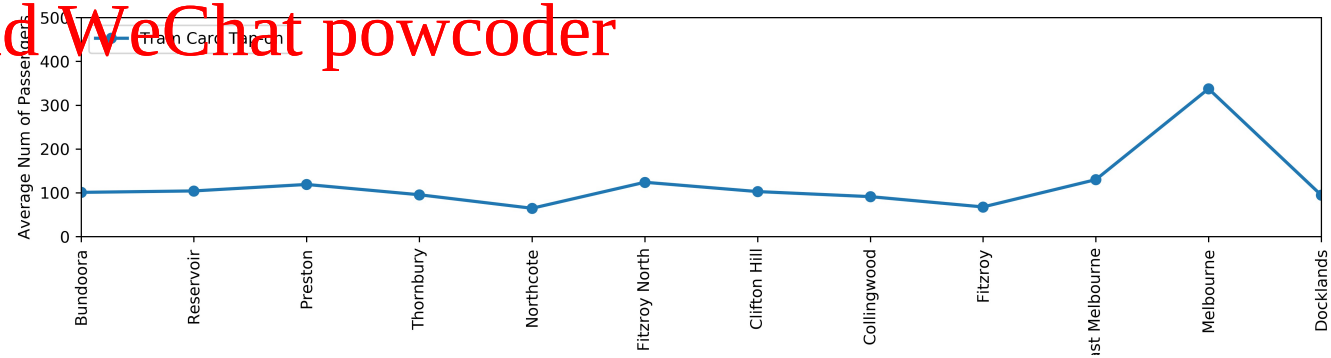
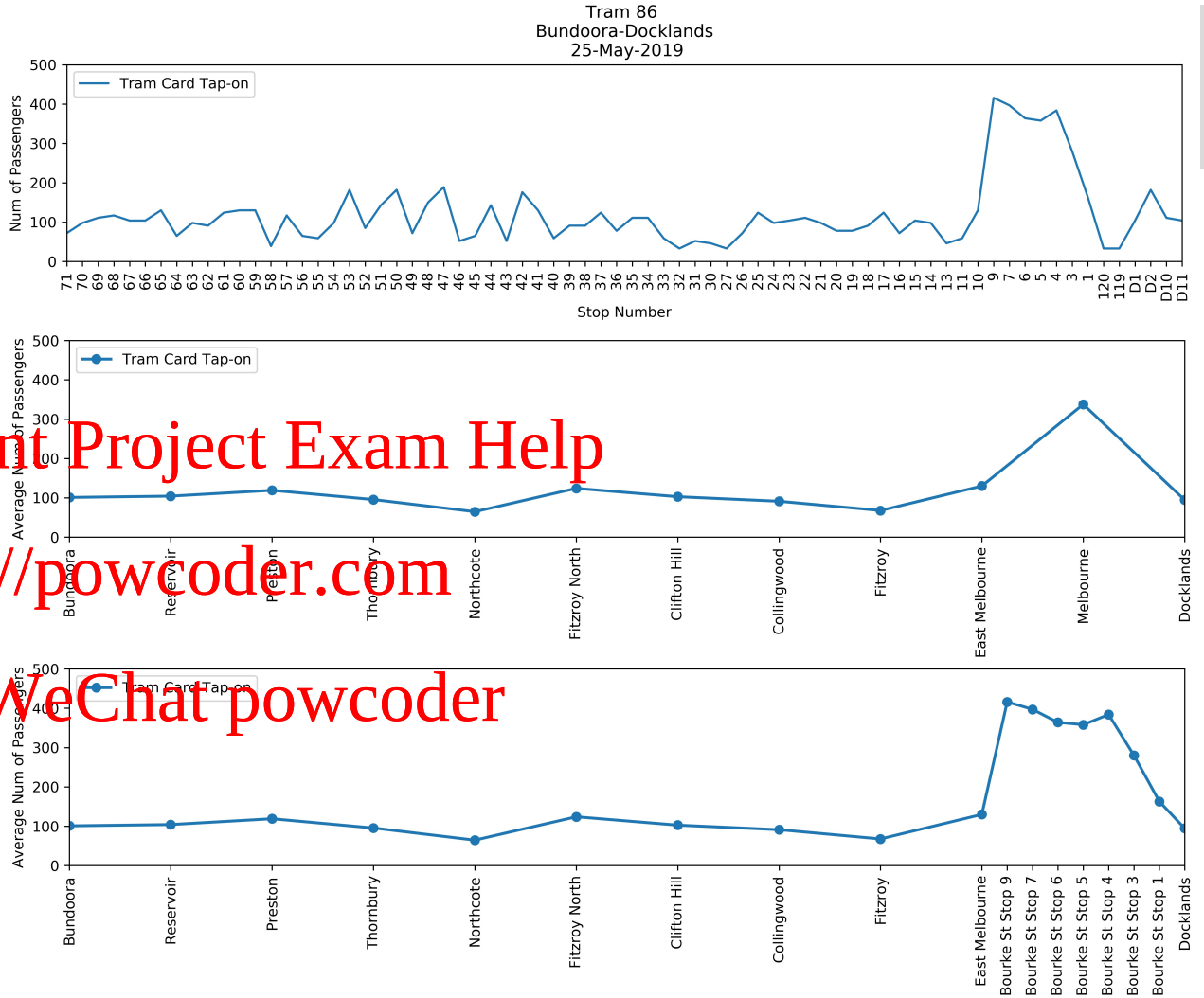


Figure 13: The tram data based on Suburb (average number of passengers for each suburb)

- Level-1 granularity

Mixed Levels of Granularity

- Spatial-based Mixed Levels of Granularity
  - Space based



- Ex: drill-down is applied to the suburb with high intake of passenger, - Melbourne CBD
- Data in Melbourne CBD is expanded to the stop level to reveal details of each stop.

Sensor Arrays

- A sensor array is a group of sensors, usually deployed in a certain geometry pattern.
- A network of distributed sensors.
- Advantage: add new dimension to the observation and hence it helps to estimate more parameters, to have better picture of the environment being observed, and improve accuracy.
- Two categories:
    1. Multiple sensors measuring the same things, and
    2. Multiple sensors measuring different things, but they are grouped together.

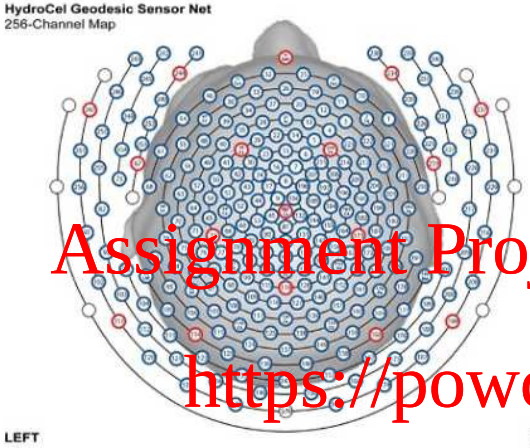
Sensor Arrays

- Multiple sensors measuring the same things
- **Why?**
  - Specialize on sensing a very specific small region
  - Get more accuracy of the results for measuring the entire region or condition

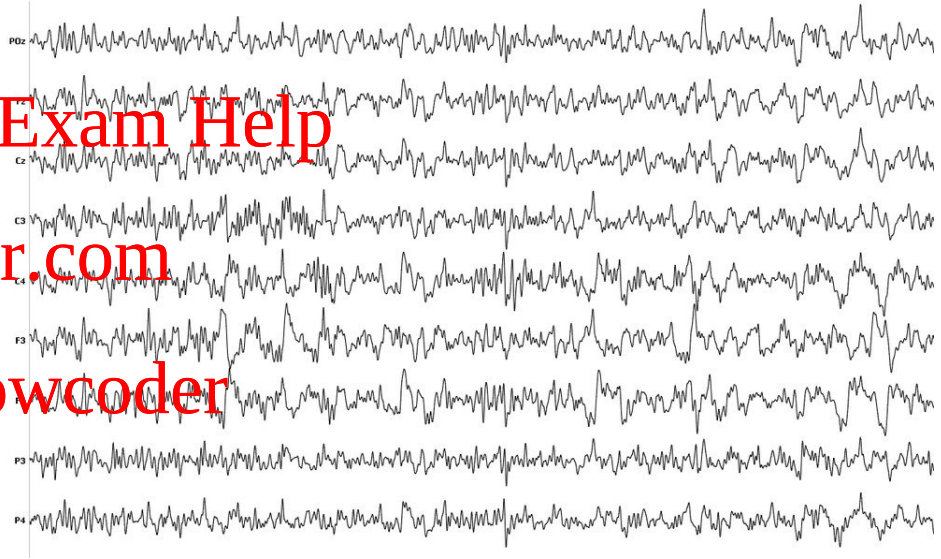
- **Example:** Multiple weather stations of a region or city measuring the air temperatures. We use three weather stations in Melbourne: Melbourne Airport, Melbourne Olympic Park and Essendon Airport.

- The aggregate of these three weather stations represent the temperature of Melbourne city. So these three weather stations are measuring the same thing i.e. the air temperature

An example of biomedical sensor arrays



Multiple sensors measuring the same things



256 electrode/sensor dense-array
EEG (electrical brain signals)

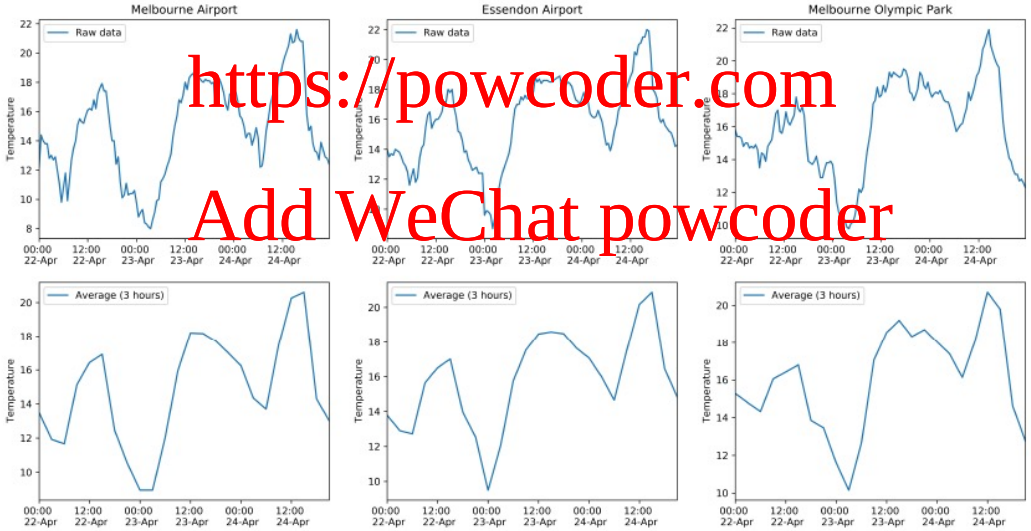
Imagesource: book: Management of Epilepsy - Research, Results and Treatment
https://www.biofeedback-tech.com/articles/2017/9/19/quantitative-eeg-and-databases-meg5l

Sensor Arrays

- Multiple sensors measuring the same things
- **Two methods to lower the granularity of sensor arrays that measure the same thing:**
  - Method 1: Reduce and then Merge
  - Method 2: Merge and then Reduce

Sensor Arrays

- Multiple sensors measuring the same things
- **Method 1: Reduce and then Merge**
  - Step 1: Reduce the granularity of each individual sensor.

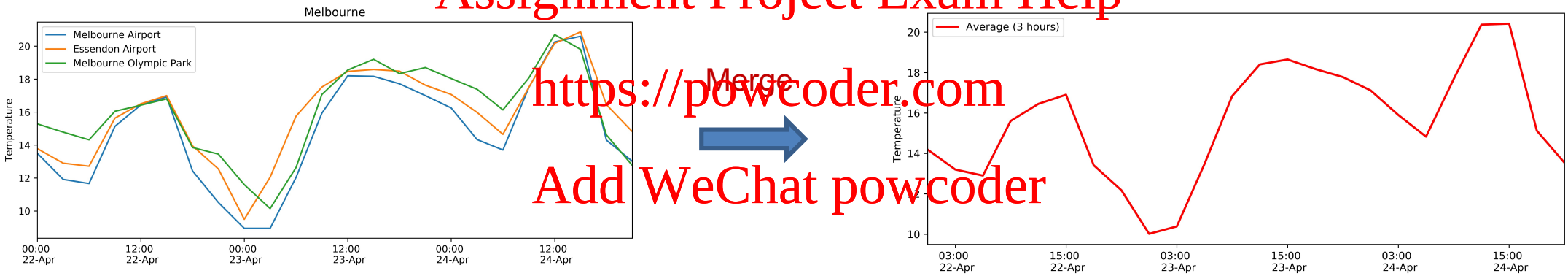


- Raw data – half-hourly data

- Average over 3 hour data

Sensor Arrays

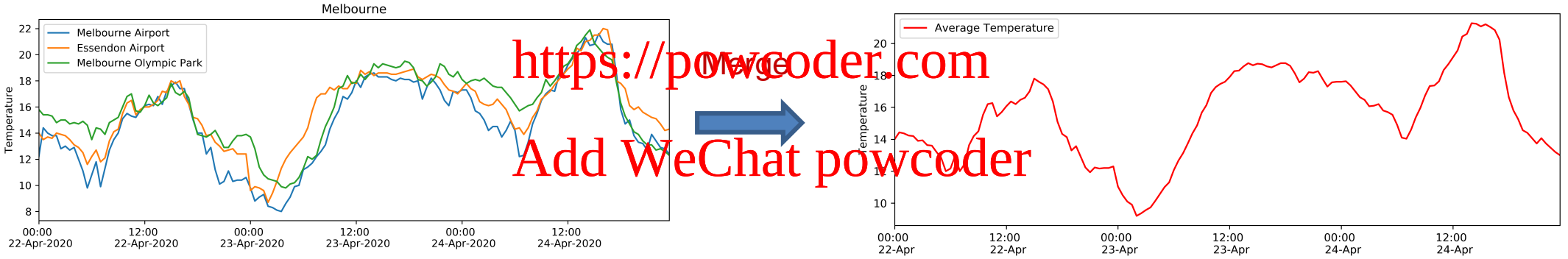
- Multiple sensors measuring the same things
- **Method 1: Reduce and then Merge**
  - Step 2: Merge reduced granularity into one data stream.



Merging: aggregate three weather stations at the 3-hour granularity, based on timestamp (e.g., using mean function)

Sensor Arrays

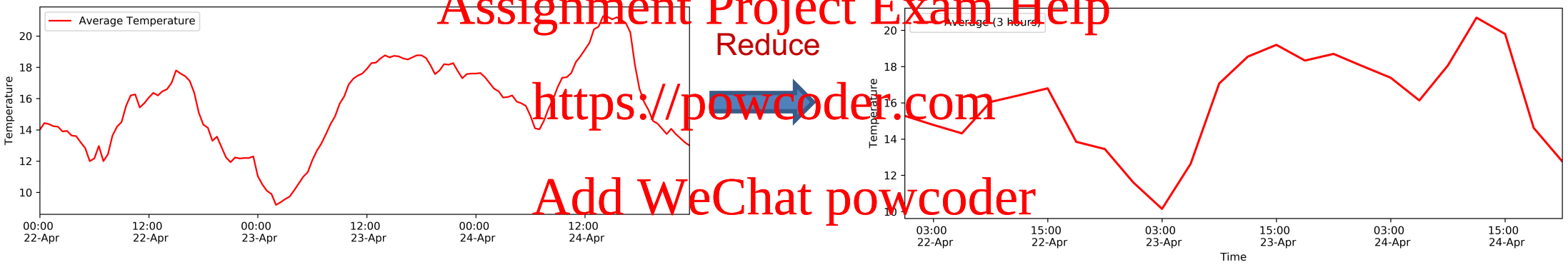
- Multiple sensors measuring the same things
- **Method 2: Merge and them Reduce**
  - Step 1: Merge the data streams first.



Merging: aggregate the raw temperature data over three weather stations

Sensor Arrays

- Multiple sensors measuring the same things
- **Method 2: Merge and them Reduce**
  - Step 2: Reduce the **granularity** of the merged data stream.



Both methods shows similar trend. For Method 2: some of details in the raw data are still preserved in the reduced granularity version

→Reason: In Method 1, details of raw data have been smoothed out in first step. In Method 2, reduced step is still based on level-0 data

Sensor Arrays

- Multiple sensors measuring different things

  - Sensors arrays can be a collection of sensors measuring different things within the same environment.

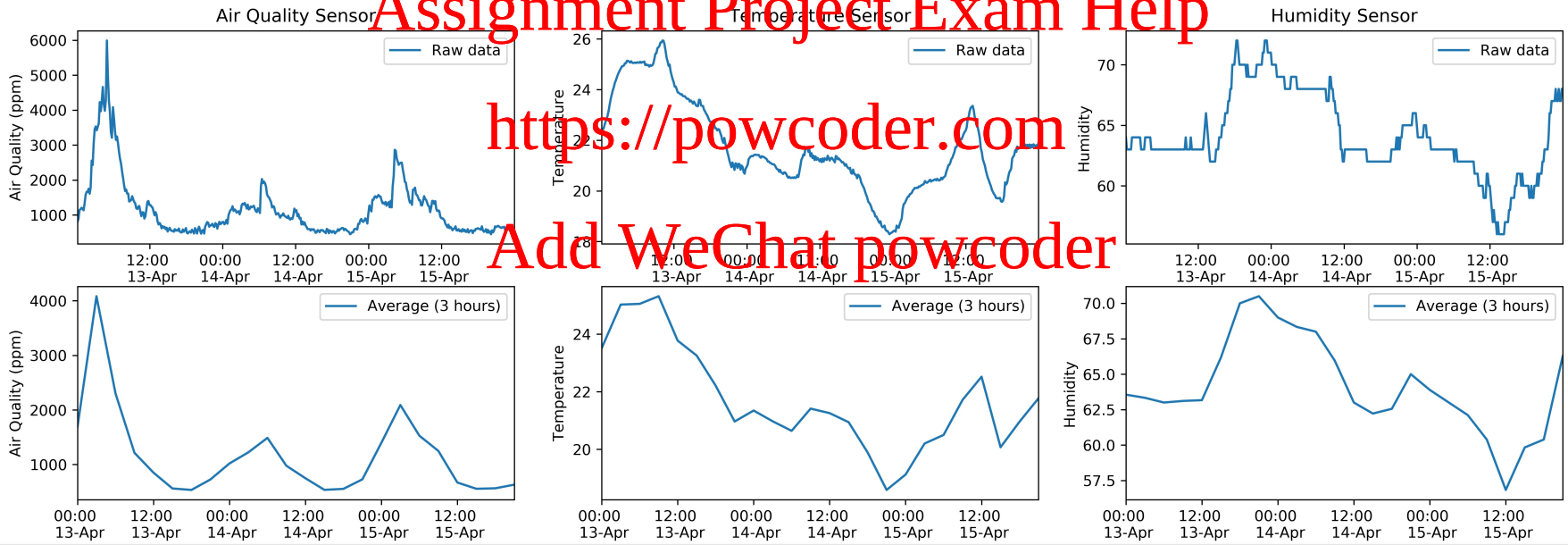
- Example: A simple indoor sensor array, containing three sensors: air quality, temperature, and humidity.

- **Two methods to lower the granularity of sensor arrays that measure the different thing:**

  - Method 1: Reduce, Normalize, and then Merge
  - Method 2: Normalize, Merge and then Reduce

Sensor Arrays

- Multiple sensors measuring different things
- **Method 1: Reduce, Normalize, and then Merge**
  - Step 1: Reduce the granularity level of each sensor's raw data.

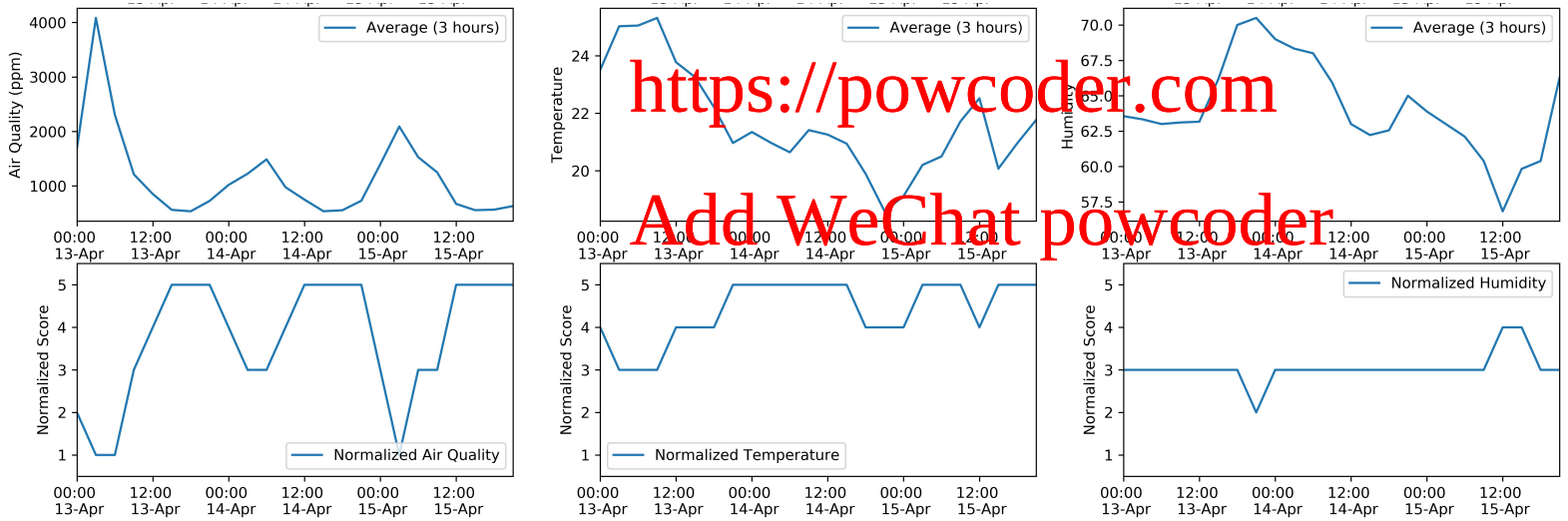


Sensor Arrays

Why do we need normalization?

- Multiple sensors measuring different things
- **Method 1: Reduce, Normalize, and then Merge**

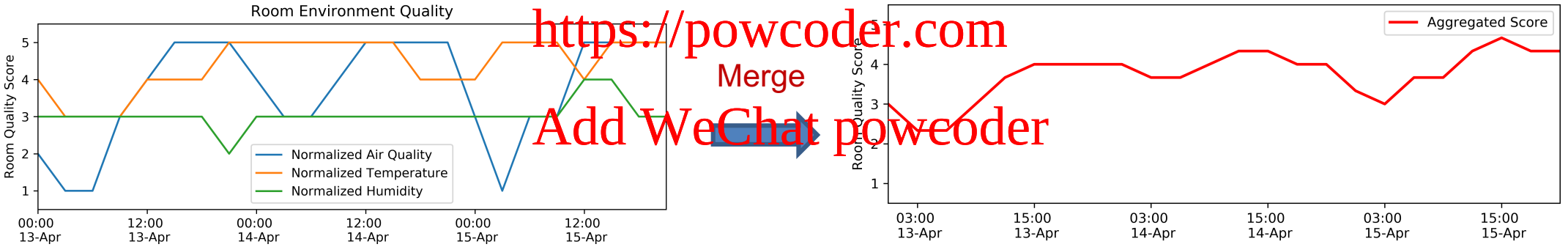
- Step 2: Normalize the reduced data of each sensor by categorizing each data into several categories (e.g, category 1 – Poor, category 5 – excellent)



Ex: 5 level of categories of room air quality:
1. Excellent: < 800 ppm
2. Good: 800 to < 1100 ppm
3. Fair: 1100 to < 1600 ppm
4. Inferior: 1600 to < 2000 ppm
5. Poor: 2000 ppm or over

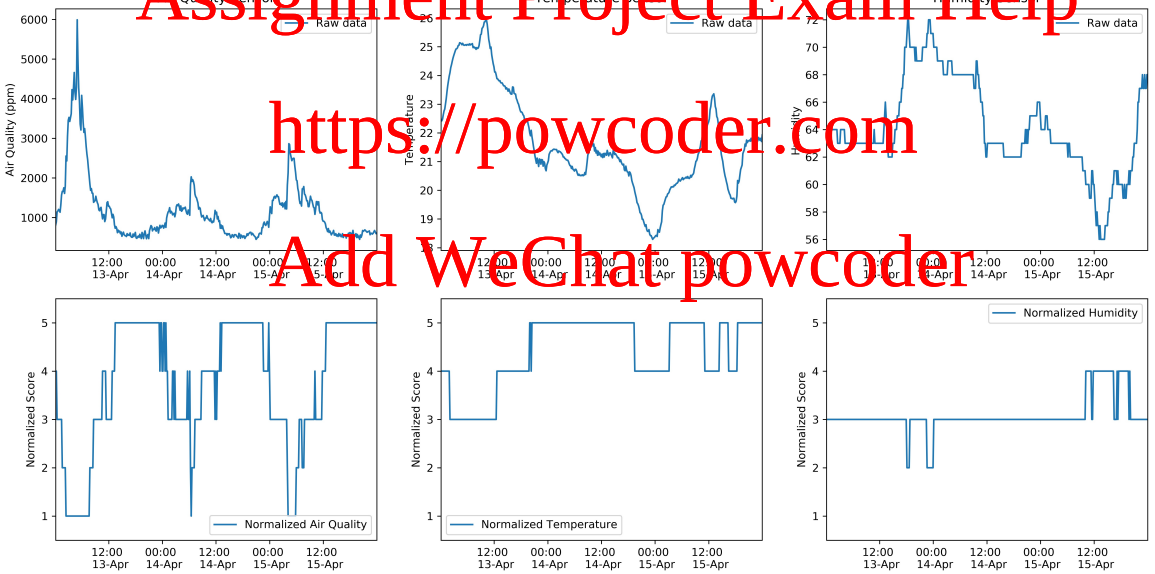
Sensor Arrays

- Multiple sensors measuring different things
- **Method 1: Reduce, Normalize, and then Merge**
  - Step 3: Merge the normalized data.



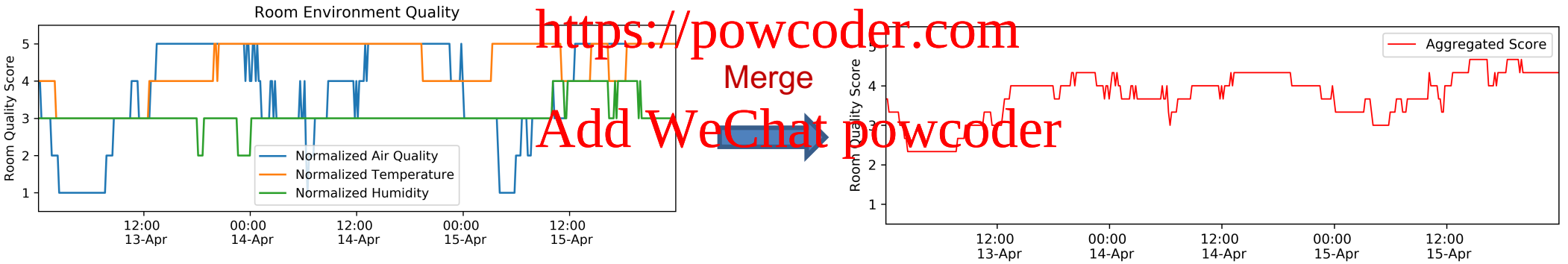
Sensor Arrays

- Multiple sensors measuring different things
- **Method 2: Normalize, Merge and then Reduce**
  - Step 1: Normalize each stream according to the categories.



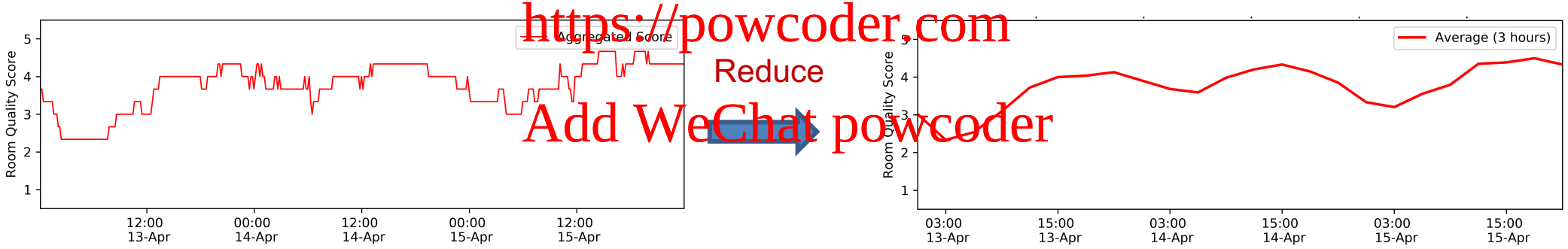
Sensor Arrays

- Multiple sensors measuring different things
- **Method 2: Normalize, Merge and then Reduce**
  - Step 2: Merge the normalized streams.



Sensor Arrays

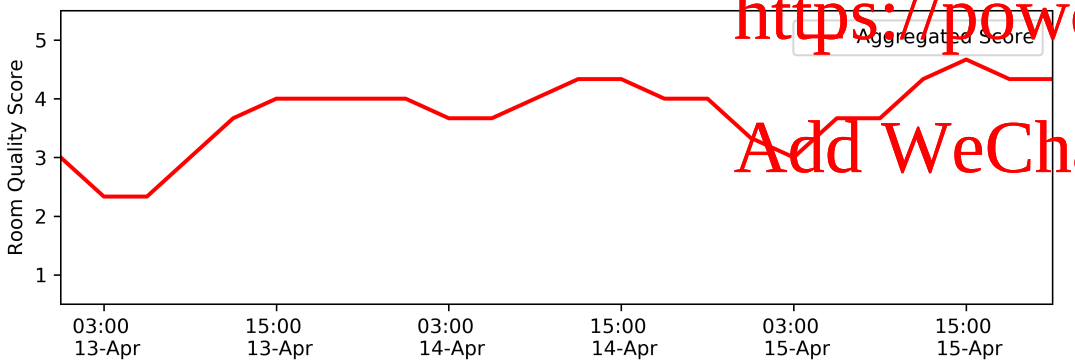
- Multiple sensors measuring different things
- **Method 2: Normalize, Merge and then Reduce**
  - Step 3: Reduce the granularity of the merged results.

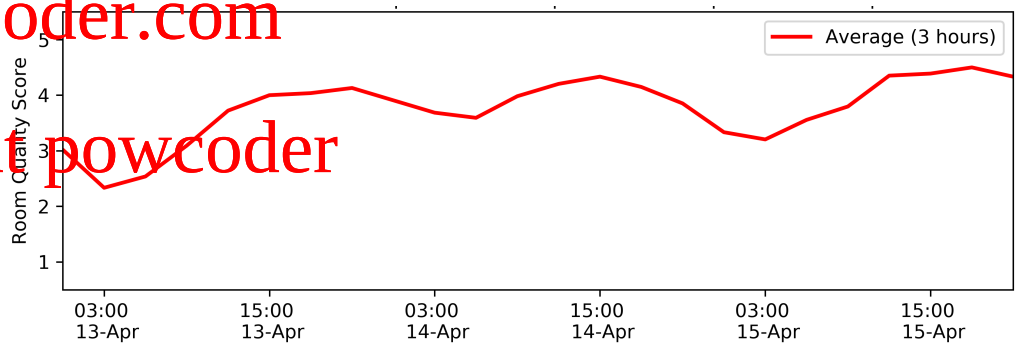


Half-hourly data data is reduced to 3-hourly data

Sensor Arrays

- Multiple sensors measuring different things
- **Method 1 vs Method 2**





Comparing Sensor Arrays

- Comparing both types of sensor arrays (e.g. multiple sensors measuring the same or different things), the multiple sensors measuring different things need to normalize the raw data, so that merging between different sensors will become possible.

- The normalisation process is to convert the raw data into a category, which binds different sensors into one common thread.

DEMO

- Multiple sensors measuring the same things
- **Method 1: Reduce and then Merge**

Pandas – resample() method – aggregate time series data for an time interval

Original

```
# Waiting for messages
for message in consumer:
    data = message.value # Json object sent by each sensor
    if data['uid'] in sensors:
        df_orig = sensors[data['uid']]['orig']
        record_time = pd.to_datetime(data['datetime'], infer_datetime_format=True)
        df_orig.loc[record_time] = pd.Series([float(data['temperature'])],['Temperature'])
        # Reduced granularity of data
        sensors[data['uid']]['reduced'] = df_orig.resample('3H').mean()
        # Merge reduced data
        df_merge = df_1.merge(df_2, left_index=True, right_index=True).merge(df_3, left_index=True, right_index=True)
        series_merged_avg = df_merge.mean(axis = 1, skipna = True)
        try:
            plotSensorGraphs(df_merge,series_merged_avg)
        except Exception as ex:
            print(str(ex))
    fig.canvas.draw() # drawing on the canvas
```

Corrected

```
# Waiting for messages
for message in consumer:
    data = message.value # Json object sent by each sensor
    if data['uid'] in sensors:
        df_orig = sensors[data['uid']]['orig']
        record_time = pd.to_datetime(data['datetime'], infer_datetime_format=True)
        df_orig.loc[record_time] = pd.Series([float(data['temperature'])],['Temperature'])
        # Reduced granularity of data
        sensors[data['uid']]['reduced'] = df_orig.resample('3H').mean()
        # Merge reduced data
        if data['uid'] == '1':
            df_1 = sensors[data['uid']]['reduced']
        elif data['uid'] == '2':
            df_2 = sensors[data['uid']]['reduced']
        elif data['uid'] == '3':
            df_2 = sensors[data['uid']]['reduced']

        df_merge = df_1.merge(df_2, left_index=True, right_index=True).merge(df_3, left_index=True, right_index=True)
        series_merged_avg = df_merge.mean(axis = 1, skipna = True)
        try:
            plotSensorGraphs(df_merge,series_merged_avg)
        except Exception as ex:
            print(str(ex))
    fig.canvas.draw() # drawing on the canvas
```

Summary

- Granularity reduction of sensor data is achieved though the windowing schemes.
- The drill-down of data streams can be assisted though multi levels of granularity which combined several granularity levels when presenting the data streams.
- Sensor arrays are multiple sensors that work together in an environment to provide users with more complete picture of the environment.

Thank You

MONASH University

Questions?