Lahore University of Management Sciences
Syed Babar Ali School of Science and Engineering, Department of Electrical Engineering
EE100 – Engineering Laboratory

Spring 2022

Project 1 – Handout

Finding the Blue Skies: Analyzing Lahore's Air Quality Data

1. OBJECTIVE

Perform basic data analysis and visualization for Lahore's air quality data.

2. BACKGROUND

Air pollution has become a major health hazard in the city of Lahore. In an effort to analyze the situation using publicly available data, our research group recently collected air quality data available from the (a) air quality monitoring station deployed at the US consulate in Lahore, (b) the monitoring stations deployed by Punjab's environment protection department (EPD), and (c) crowd-sourced sensor network available from PurpleAir. This project will have you conduct some preliminary data analysis with this data. For information about our results, please refer to our blog available here.

3. DATA

Publicly available data of daily PM2.5 concentrations was collected from various sensors across Lahore. The location of these sensors (latitude / longitude) are available in the accompanying csv file (on LMS) *locations.csv*.

S. No.	Sensor ID	Owner
1	USEmbassy	AirNow
2	RenkeLUMS	LUMS
3	PAirGardenTown	PurpleAir
4	PAirDHAPhase2	PurpleAir
5	PAirAnarkali	PurpleAir
6	PAirHarbanspura	PurpleAir
7	PAirDefenceChowk	PurpleAir
8	PAirIqbalTown	PurpleAir
9	PAirAkbarChowk	PurpleAir
10	PAirTownship	PurpleAir
11	MeTStation	EPD
12	Dental College	EPD
13	EPD Gulberg	EPD

Table 1. Air quality sensors used in this project.

The daily concentrations of PM2.5 recorded at these locations (in µg/m^3) from January 1, 2019 to January 20, 2022 is available in the accompanying file *AirQualityData.csv*. Note that there are swathes of missing data from each location at different durations. Missing data in the file is indicating by an entry of "-1".

4. PROJECT TASKS

In addition to this handout of the project, you have been provided with two csv files that contain locations of the sensors and their corresponding daily PM2.5 measurements. Note that upon completion of the tasks, you must submit your MATLAB code and a report describing all your steps and rationale for any design choices that were made in the process. The exact tasks in the narrative below are marked by bullet points and in the color blue. The text without bullet points is relevant information that you may find useful in understanding some of the whats and the whys.

Task 1: Displaying location of sensors

- Use the latitude and longitude data in the given csv file to create a map of where the sensors are located in the city of Lahore. You are required to produce something like the figure below. For the purpose, read through the documentation available **here** and use functionalities listed on the page to write the required MATLAB code.

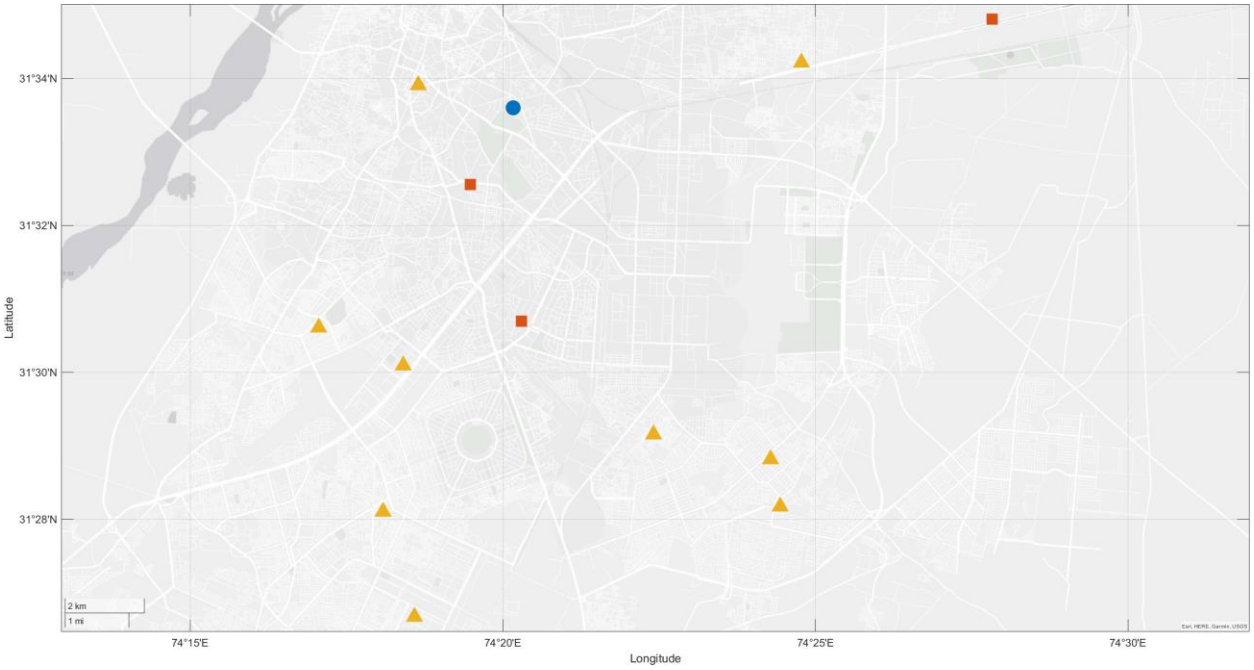

Fig. 1. Expected output of Task 1: A geoscatter plot of the sensor locations

This type of a plot is called a geoscatter plot. Note that the *geoscatter* plot given above uses three different types of markers. The circle indicates the location of the US embassy, the squares indicate the locations of the EPD stations, while the triangles indicate the locations of the cheap crowd-sourced sensors (Rows 2 – 10) of Table 1.

Task 2: Calculating Distances

Use the given latitude and latitude to compute the pair-wise distance (in kilometers) of any given sensor from all other available sensors. Since there are 13 sensors in total, this would require you to compute a total of 78 unique distances (13 times 12, divided by 2). In order to compute the distance between two points whose lat/long are given, the *haversine* formula (based on calculations over a sphere) is often used.

An example resource that gives details of this formula can be found **here**. This link also includes a web tool that computes for you the distance between two points that you input. This will be helpful in debugging your MATLAB script.

- Write MATLAB code that computes the distance between all pairs.
- Write MATLAB code that plots the distance of each station from the US embassy station in the form of a bar graph. For plotting bar graphs, the documentation **here** will be helpful – look through the section "Specify Categorical Data' on this link to find the most relevant information. Your output should look something like the figure below.

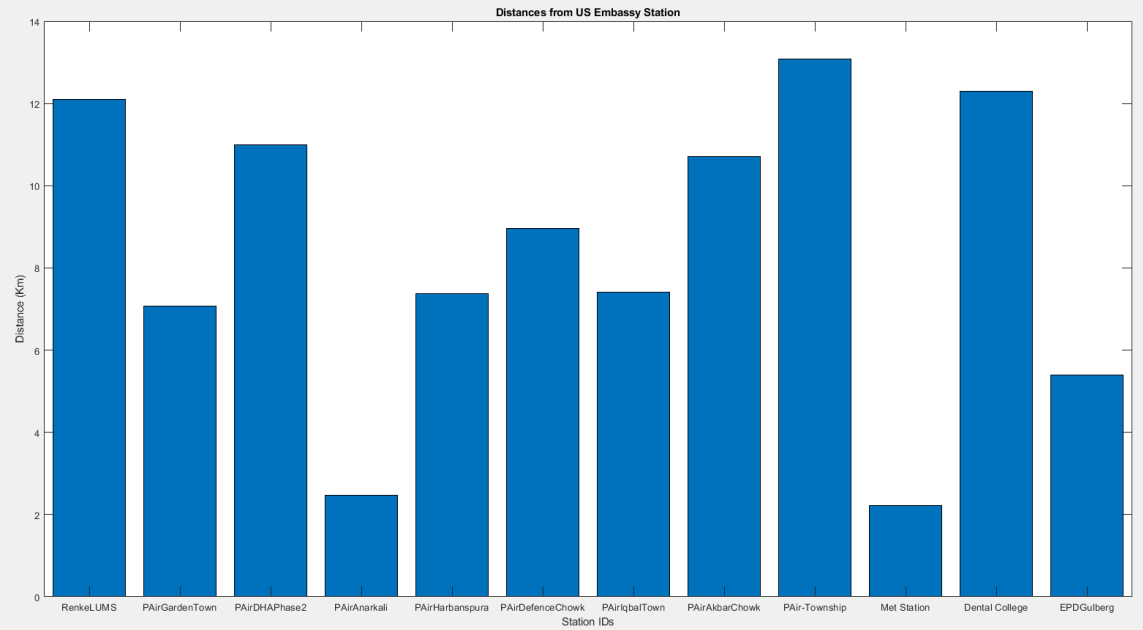

Fig. 2. Expected output of Task 2: A bar graph of distances from US embassy station.

Task 3: Computing Correlation Coefficients

Recall from Lab 9 that correlation is a measure that captures the amount of similarity between two signals. In this task we will attempt to use correlation coefficient between the reported values of different sensors to determine the data integrity of different sensors.

The US embassy station is reference grade and may be considered the most reliable of sources. Consequently, this could be considered as a gold standard against which all the other 12 sensing stations could be compared. Write MATLAB code to

- Extract the air quality data (form the relevant csv file) from August 1 2021 till January 18, 2022.
- Compute the pair-wise correlation coefficient between the US embassy data and *all* other 12 sensing stations. When computing this correlation, note that you will need to remove those dates from your computation in which either one of the two readings are missing.
- Plot a bar graph of the correlation coefficients, similar to the one in Task 2.

- You may not be able to compute the correlation with the EPDMetStation and DentalCollege stations. Explain the reasoning for this in your report.

You will find that the EPD Gulberg data has the smallest correlation with the data from the US embassy, whereas all cheap sensors have a correlation coefficient greater than 0.9. The EPD Gulberg station is situated more than 5 Km away from the US embassy, so this distance may have a role to play in the small correlation. However, were distance the significant factor in this observation, one would not have observed high correlations for cheap sensors that are even more than 12 Km away (e.g., the PurpleAir sensor in Township). *This leads us to doubt the integrity of the data being reported by the EPD Gulberg station*.

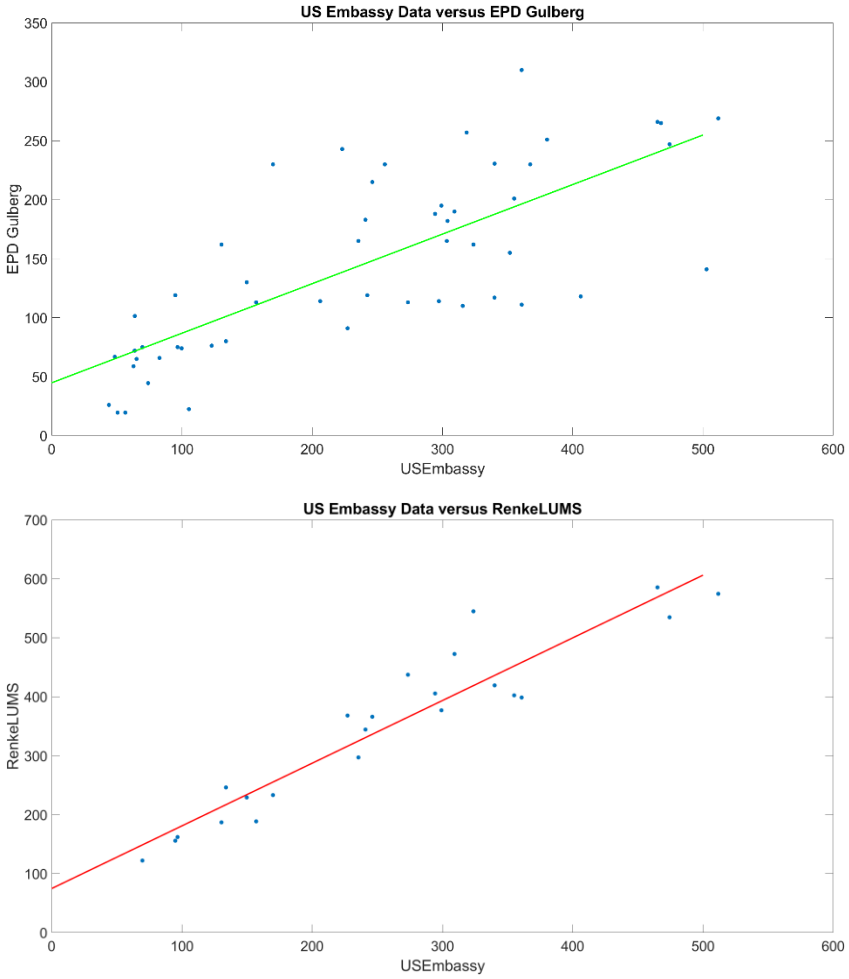

Fig. 3. US embassy data versus EPD Gulberg (top) and Renke LUMS (bottom). The linear trend in the bottom figure is much more visible than the top.

What does the correlation coefficient represent? One interpretation is that it quantifies the linear relationship between two data streams / signals. To see this, look at Fig. 3. For every date for which both stations had data available, the top figure plots the US embassy data on the x-axis and the corresponding EPD Gulberg data on the y-axis. The bottom figure does the same for the US embassy data versus that obtained from our sensor at LUMS. Clearly, the data in the bottom figure follows a stronger linear

relationship than the one at the top. This is indicated by the higher correlation coefficient between the US embassy and Renke LUMS as compared to that between US embassy and EPD Gulberg.

Task 4: Calibrating the Sensor Readings

The sensors 2 through 10 in Table 1 are relatively cheap sensors (approximate cost is around US $300) versus the reference grade stations that typically cost up to US $100,000 to install and maintain. The use of cheap sensors for making policy decisions is debatable since their measurements are often subject to errors. Nevertheless, there appears to be an appetite for using hybrid solutions where a few reference grade stations in a city are interspersed with many cheap measurement systems, with the measurements of the former used as beacons for calibrating errors in the latter. In this task, we will have you carry out the simplest calibration which is linear.

To understand linear calibration, refer to Fig. 3. When comparing the US embassy data versus Renke LUMS, we used a first order (linear) polynomial fitting depicted with the red line. This is done through the `polyfit` command (see its documentation to figure out how it works). The polynomial fitting gives an output

```
[1.0628    74.7406]
```

This means that the linear approximation is given by

```
RenkeLUMSReading = 1.0628 x USEmbassyReading + 74.7406
```

If we ***assume*** that the US embassy output is *equal* to the actual PM2.5 concentration at LUMS, then the LUMS sensors may be calibrated as follows:

```
RenkeLUMSReadingCalibrated = (RenkeLUMSReading - 74.7406)/1.0628
```

- Given the information above, devise a mechanism for calibrating the data for sensor locations in rows 2 through 10 of Table 1. Write MATLAB code that implements this calibration.
- Clearly outline in your report the details of your mechanism, as well as the reason for your design choices.
- Clearly outline the shortcomings of your mechanism. What would you have needed to mitigate the effect of some of those shortcomings?

Task 5: Estimating the missing values at LUMS

From the data, you will observe that the RenkeLUMS sensor came online on December 24, 2021 meaning that we do not have any air quality at LUMS before this date. We notice however, due to the underlying physics of the phenomenon, that the measurements made at other locations in Lahore may give us an *estimate* of the air quality at LUMS. One of the simplest methods is *inverse distance weighting*, which you will be required to implement in this task. On any given date, the method will form the estimated AQI value at LUMS by computing a weighted average of the available readings of the other sensors in the city with the weights chosen to be inversely proportional to the corresponding distances. In other words, the readings from sensors closer to LUMS should be weighted more than the ones farther away. Mathematically, the estimated value at any given date is given as

$$Y = \frac{\sum_i w(i)X(i)}{\sum_j w(j)}$$

where the summation is over the set of all sensors whose measurement is available on that given day, *X(i)* is the reading of the i^{th} sensor in that set, and *w(i)* is the inverse of the distance between the location of the i^{th} sensor and LUMS (calculated in Task 2).

- Write MATLAB code that estimates the PM2.5 concentration at LUMS from August 1, 2021 till December 23, 2022 using inverse distance weighting. You may use the calibrated data from Task 4 for the purpose. Since we have established that the EPD stations' data may not be reliable, you may not use them for the estimation. Plot the estimated concentrations against the dates.
- Is there any way that would allow you to determine how accurate the estimation method is? Discuss in your report.

Project Report. Please submit a project report/documentation that provides the information requested in the task description. The report should also include a listing of the individual contributions of each team member.

CHECKPOINTS AND TIMELINE

S.No	Tasks	Deadline	Instructions
1	Project selection and team formation	**L3: Sat, April 9, 11:55 pm** **L4: Tue, April 12, 11:55 pm**	Send an email to Nouman Arshad (nouman.arshad@lums.edu.pk) indicating your choice of project and a list of team members. **Max. No. of Team Members:** 3 **Min. No. of Team Members:** 2
2	**Checkpoint 1:** Complete Task 1,2,3	**L3: Fri, April 15, 5:00 pm** **L4: Mon, April 18, 5:00 pm**	You will be assigned a dedicated TA who will track your progress. Show your code and results to the TA in the lab/office hours.
3	**Checkpoint 2:** Complete Task 4,5	**L3: Fri, April 22, 5:00 pm** **L4: Mon, April 25, 5:00 pm**	Show your code and results to your TA in the lab/office hours.
4	Final code and report submission with all 5 tasks completed	**L3: Tue, April 26, 5:00 pm** **L4: Fri, April 29, 5:00 pm**	Submit your MATLAB code and any stipulated documentation / project report.