# CS685A: Assignment 1

## Analysis of Covid, Vaccination and Census Data

## Problem Description

- Can be found in the Assignment file.

## Requirements:

- Python3 should be present on your system and should be accessible using the command `python3`

## Running the Project:

- To run the complete project in one go, execute `assign1.sh`

```
./assign1.sh
```

- To run a certain segment, use the following scripts:
    1. `edge-generator.sh` : To solve Q1 and Q2
    2. `case-generator.sh` : To solve Q3
    3. `peaks-generator.sh` : To solve Q4
    4. `vaccinated-count-generator.sh` : To solve Q5
    5. `vaccination-population-ratio-generator.sh` : To solve Q6
    6. `vaccine-type-ratio-generator.sh` : To solve Q7
    7. `vaccinated-ratio-generator.sh` : To solve Q8
    8. `complete-vaccination-generator.sh` : To solve Q9
- The individual scripts need to be executed in the order in which they are mentioned, as the output generated by one script maybe needed for the execution of the further scripts.

## Project Structure:

### Directories:

- out/ : Stores the required output json and csv files.
- data/ : Modified and Refined data files used in the project.
- meta/ : Stores the meta data (a set of json files) generated by the execution of scripts. The contents inside this folder ***should not*** be modified when the scripts are being executed. The folder can be deleted before beginning the execution of `assign1.sh` or `edge-generator.sh`.
- base/ : Contains the initial unmodified data files.
- update/ : Redundant copy of `data/`, used for debugging and comparisons.

- util/ : Contains python script(s) that have snippets used to clean and refine the original data.

## Code Files:

### Shell Scripts:

- Already described here.

### Python Scripts:

- meta.py : Converts the csv files to json format for use by the scripts later on. It also finds issues in multiple data sources (district pairs with edit distances < x, districts contained in another, etc.)
- q1.py : Solves Question 1. Uses `neighbor-districts.json` and `district_wise.csv` to generate `neighbor-districts-modified.json`. See **Explaination 1**.
- q2.py : Creates au undirected graph from `neighbor-districts-modified.json`. This graph is printed in edge list format `edge-graph.csv`. See **Explaination 2**.
- q3.py :
- q4.py :
- q5.py :
- q6.py :
- q7.py :
- q8.py :
- q9.py :

# Explaination:

- The new json districts does a deep merge of the districts, i.e., if A, B and C are connected such that, originally A is a neighbour of B and B is a neighbour of C but A and C are not connected directly, now if B is to be removed, A and C will be reported as neighbours in the final answer.
- DFS is used to generate the edge list. During DFS, parent history is maintained to prevent reporting duplicate edges. Edge list is printed directly (no header line is printed in the csv).