# DiscSim

This code was written for [`CEGIS`](https://www.cegis.org/) (Center for Effective Governance of Indian States), an organization that aims to help state governments in India achieve better development outcomes.

An important goal of CEGIS is to improve the quality of administrative data collected by state governments. One way in which this is achieved is to re-sample a subset of the data and measure the deviation from the original samples collected. These deviations are quantified as 'discrepancy scores', and large discrepancy scores are flagged for intervention by a third party.

Often, it is not clear what re-sampling strategy should be used to obtain the most accurate and reliable discrepancy scores. The goal of this project is to create a simulator to predict discrepancy scores, and the statistical accuracy of the discrepancy scores, for different re-sampling strategies. This repository will be populated with python scripts and jupter notebooks to implement the simulator. However, no data will be made public as it is sensitive data collected by state governments in India.

Glossary

1. Subordinate: collector of original samples

2. Supervisor: collector of secondary samples (used to calculate discrepancy score)

To use the code in this repository, clone the repository and set up the anaconda environment using the `environment.yml` file. You can follow instructions [here](<https://conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html#creating-an-environment-from-an-environment-yml-file:~:text=%2D%2Dhelp.-,Creating%20an%20environment%20from%20an%20environment.yml%20file,-%EF%83%81>).

A good starting point is to play with the jupyter notebooks in the `Scripts` folder:

1. `binomial\_confidence.ipynb` models discrepancy scores of binary variables as a binomial distribution, and uses the model to predict the number of samples required to confidently classify subordinates into categories based on discrepancy score.
2. You can read more about binomial random variables here: <https://online.stat.psu.edu/stat800/book/export/html/658#:~:text=A%20binomial%20random%20variable%20counts,(a%20fixed%20sample%20size)>
3. You can read more about cumulative distributive functions here: <https://www.probabilitycourse.com/chapter3/3_2_1_cdf.php>
4. `Sample EDC data analysis.ipynb` uses dummy data from the education sector to calculate discrepancy scores using various methods, and asses the statistical significance of those discrepancy scores.
5. `sampling\_strategy.ipynb` compares random sampling and ranked sampling of beneficiaries for discrepancy score calculation, using two different metrics - accuracy of discrepancy score, and efficacy of catching 'worst offender' subordinates.

To run a jupyter notebook, navigate to the directory containing the cloned repository in the terminal/command prompt. Type in the following commands (make sure you have set up the environment `discsim` using the `environment.yml` file as discussed above):

```python

>>>conda activate discsim

>>>jupyter notebook

```

This should open a browser tab in which you can navigate to the `Scripts` folder and click on any of the jupyter notebooks above to run them. Run the cells in order and make changes in the code to explore (for an introduction to jupyter notebooks, see [this page](https://realpython.com/jupyter-notebook-introduction/) (note that you don't need to install jupyter - it is already installed in the environment `discsim`).