Project Report for Summer Undergraduate Research Award 2023

Data Envelopment Analysis for the Performance of Various Geographical Regions in Fighting COVID-19

Facilitator:Prof.Aparna Mehra

Department Head:Prof.Aparna Mehra

Authors

Rakshitha
2021MT10904
mt1210904@iitd.ac.in

Ayush Gupta
2021MT10697
mt1210697@iitd.ac.in

Contents

1 Notation

- **DMU**: Decision Making Unit
- $\mathbf{x_{ij}}$: ith input of the DMU_j
- $\mathbf{y_{rj}}$: rth output of the DMU_j
- $\mathbf{S_i^-}$:slack associated with ith input of DMU_o
- $\mathbf{S_r^+}$: slack associated with rth output of DMU_o
- $\boldsymbol{\alpha}_{\mathbf{ij}}$: incremented ith input of the DMU_j
- $\boldsymbol{\beta}_{\mathbf{rj}}$: incremented rth output of the DMU_j
- **n**: Number of DMUs
- $\boldsymbol{\theta}_{\mathbf{o}}$: Efficiency of DMU_o
- $\widetilde{\mathbf{x_{ij}}}$: ith random input of the DMU_j
- $\widetilde{\mathbf{y_{rj}}}$: rth random output of the DMU_j
- $\boldsymbol{\alpha}$: confidence level

2 Introduction

The November 2019 outbreak of COVID-19 marked a pivotal moment in the history of public health and global healthcare infrastructure. The emergence of this novel coronavirus sparked an unprecedented surge in hospital admissions around the world, straining medical facilities and demanding immediate, effective responses. The symptoms associated with the virus, such as fever, cough, difficulty breathing, and other respiratory distress, necessitated intensive care unit (ICU) treatment for many patients. However, the global landscape of healthcare systems and the political dynamics of various nations resulted in diverse and often contrasting responses to this unprecedented crisis, both in terms of demographics and economics.

We aim to delve into the multifaceted response to the COVID-19 pandemic, shedding light on how different geographical regions, and governments reacted to the crisis. Factors such as the existing health infrastructure, resource allocation mechanisms, and the availability of essential resources played a crucial role in shaping the strategies employed by different geographical regions. In our nation, India, for instance, the central government took charge of spearheading the campaign against COVID-19, while state governments were granted significant autonomy in resource allocation and handling the situation within their own jurisdictions. The absence of recent experiences dealing with a pandemic of this magnitude, coupled with the urgency of the situation, often resulted in disorganized and, at times, ineffectual responses. These experiences have left us with a wealth of invaluable lessons to be learned.

The COVID-19 pandemic has not only exposed the vulnerabilities within our healthcare systems but has also led to significant advancements in research and understanding of pandemics. Mathematical studies of the pandemic's dynamics and the optimization of resource utilization during the COVID era have become critical areas of study, offering insights into how we can better prepare for and respond to future healthcare crises. As we explore the various aspects of this global health emergency in this report, we will gain a deeper understanding of the challenges and opportunities it presented, with the hope that these findings will inform and guide future strategies for pandemic preparedness and response.to the lack of recent similar experiences and the short reaction time, the bulk of responses were disorganised and ineffectual, leaving us with a wealth of valuable lessons. The pandemic resulted in several research advancements. Some of them were based on the mathematical study of the pandemic's dynamics and the optimization of resource use during the COVID era.

2.1 Data Envelopment Analysis (DEA)

Data Envelopment Analysis (DEA) is a powerful mathematical method utilized for the evaluation of the relative efficiency of Decision-Making Units (DMUs) that share similarities. These DMUs can represent organizations, or institutions. DEA assesses their performance using a set of inputs and outputs categorized as performance indicators. The primary goal is to identify and analyze the efficiency frontier, a boundary that distinguishes efficient DMUs from inefficient ones based on their utilization of resources.

DEA has found widespread applications in various sectors, with particular popularity in the banking industry. Its appeal stems from its ability to handle multiple outputs of varying types, making it suitable for complex and multifaceted performance evaluations. Unlike some other efficiency measurement methods, DEA does not require a priori knowledge of a production function, which makes it versatile and adaptable to different contexts.

Beyond the banking sector, DEA has been successfully employed in diverse fields such as insurance, agriculture, supply chain management, transportation, public policy analysis, and mutual funds. Its adaptability and ability to assess the performance of various DMUs make DEA a valuable tool for decision-makers and analysts seeking to enhance resource allocation and operational efficiency.

2.2 Inverse Data Envelopment Analysis (InvDEA)

While traditional Data Envelopment Analysis (DEA) calculates the technical efficiency of each DMU when the inputs and outputs are known, InvDEA operates in reverse. Inverse Data Envelopment Analysis is a mathematical method used to estimate either the output levels that correspond to changing input levels or, conversely, the input levels that correspond to changing output levels while maintaining a predetermined level of efficiency. In essence, InvDEA helps in determining the necessary resource levels to achieve a desired level of efficiency, making it a valuable tool for decision-makers and analysts seeking to optimize resource allocation.

As a part of this project, we try to utilize the DEA and InvDEA methods to analyse the working of the health sectors of various geographical domains in the COVID-19 times, and also suggest changes that could be incorporated by the regions to better their performance.

3 Mathematical Modelling and Setup of the Problem

3.1 Terminology

Before introducing the basic models of DEA, we need to define some key terms that are essential for understanding the concept and methodology of efficiency and productivity measurement.

1. A Decision Making Unit (DMU) is an entity that uses input(s) to produce output(s). A DMU can be an individual, a group, an organization, or any unit that performs some activities or tasks.

2. An input is a parameter that the DMU would wish to reduce if possible. Inputs can be seen as the investments that a DMU makes. Undesirable outcomes(negative outputs) could also be treated as inputs.

3. An output is a parameter that the DMU would wish to increase if possible. Outputs are generally the final positive outcomes that can be extracted from a DMU.

4. Benchmarking is a systematic procedure involving the assessment of a DMU's performance in relation to other DMUs or established benchmarks and best practices. It serves the purpose of discerning the strengths and weaknesses of a DMU, offering valuable insights for enhancement. Furthermore, benchmarking facilitates the ranking, rating, or classification of DMUs based on their efficiency scores or other pertinent performance indicators.

5. We define efficiency = $\frac{\text{virtual output}}{\text{virtual input}}$, which is the ratio of weighted sums of outputs to that of inputs. The weights are unknown parameters that are decided by the model. Efficiency serve as a quantitative measure of how well a DMU utilizes its inputs to produce its outputs.

6. A production possibility set is the collection of all possible input-output combinations. Given resource and technology based constrainsts of a DMU, its production possibility set is decided.

3.2 Input Parameters

The input parameters that we have chosen for our analysis are as follows:

1. Number of diagnostic tests conducted.

2. The number of affected individuals, representing the count of infections.

3. Health infrastructure - including the number of hospitals, beds, ICU units etc.

4. Health professionals, encompassing doctors and nurses.

5. Population density.

(Please note that we divided all input parameters by the population density of the respective geographical region, as it makes more sense to account for the amount of resources/population when analysing the data)

3.3 Output Parameters

The output parameters that we have chosen for our analysis are as follows:

1. Recoveries, which encompass both home recoveries and hospital recoveries.
2. The number of deaths, a negative output parameter.

Note that the two output parameters are interlinked (recoveries = infections - deaths), and hence just one of them is used in the final model.We now move on to the setup of the mathematical model for the analysis.

3.4 BCC Model (Output-Oriented)

$$
\begin{aligned}
&\max\ \eta_B \\
&\text{subject to:} \\
&\sum_{j=1}^{n} \lambda_j x_{ij} \le x_{io}, \qquad i = 1, 2, \ldots, m \\
&\sum_{j=1}^{n} \lambda_j y_{rj} \ge \eta_B y_{ro}, \quad r = 1, 2, \ldots, s \\
&\lambda_j \ge 0, \qquad j = 1, 2, \ldots, n \\
&\sum_{j=1}^{n} \lambda_j = 1
\end{aligned} \tag{1}
$$

3.5 Inverse-BCC Model (Output-Oriented)

$$
\begin{aligned}
&\max \sum_{j=1}^{s} w_j * \beta_{jo} \\
&\quad \text{subject to:} \\
&\quad \sum_{j=1}^{n} \lambda_j x_{ij} \leq (x_{io} + \alpha_{io}), \qquad i = 1, 2, \ldots, m \\
&\quad \sum_{j=1}^{n} \lambda_j y_{rj} \geq \eta_B (y_{ro} + \beta_{ro}), \quad r = 1, 2, \ldots, s \\
&\quad \lambda_j \geq 0, \qquad j = 1, 2, \ldots, n \\
&\sum_{j=1}^{n} \lambda_j = 1
\end{aligned}
\tag{2}
$$

3.6 SBM Model

$$
\begin{aligned}
&\min \rho = \frac{1 - \frac{\sum_{i=1}^{m} \frac{s_i^-}{x_{io}}}{m}}{1 + \frac{\sum_{j=1}^{s} \frac{s_r^+}{y_{ro}}}{s}} \\
&\text{subject to:} \\
&\sum_{j=1}^{n} \lambda_j x_{ij} = x_{io} - s_i^-, \qquad i = 1, 2, \ldots, m \\
&\sum_{j=1}^{n} \lambda_j y_{rj} = y_{ro} + s_r^+, \qquad r = 1, 2, \ldots, s \\
&\lambda_j \geq 0, \qquad j = 1, 2, \ldots, n \\
&s_i^- \geq 0, \qquad i = 1, 2, \ldots, m \\
&s_r^+ \geq 0, \qquad r = 1, 2, \ldots, s
\end{aligned}
\tag{3}
$$

After converting it into an LP program, we get the following model with transformed decision variables:

$$
\begin{aligned}
\min \theta_0 = t - \frac{\sum_{i=1}^{m} \frac{S_i^-}{x_{io}}}{m} \\
\text{subject to:} \\
t + \frac{\sum_{j=1}^{s} \frac{S_r^+}{y_{ro}}}{s} = 1 \\
\sum_{j=1}^{n} \Lambda_j x_{ij} = t x_{io} - S_i^-, \qquad & i = 1, 2, \ldots, m \\
\sum_{j=1}^{n} \Lambda_j y_{rj} = t y_{ro} + S_r^+, \qquad & r = 1, 2, \ldots, s \\
\Lambda_j \geq 0 \qquad & j = 1, 2, \ldots, n, \\
t \geq 0, \\
S_i^- \geq 0, \qquad & i = 1, 2, \ldots, m \\
S_r^+ \geq 0, \qquad & r = 1, 2, \ldots, s
\end{aligned}
\tag{4}
$$

3.7 Inverse-SBM Model

Suppose the inputs of DMU_o are increased from from x_o to $\alpha_o = x_o + \Delta x_o$ ($\Delta x_0 \geq 0$) and the expected output is $y_o + \beta_o$, then the inverse SBM model is as follows:

$$
\begin{aligned}
\max \beta_o = (\beta_{1o}, \beta_{2o}, \ldots, \beta_{so}) \\
\text{subject to:} \\
t + \frac{\sum_{j=1}^{s} \frac{S_r^+}{\beta_{ro}}}{s} = 1 \\
\sum_{j=1}^{n} \Lambda_j \alpha_{ij} = t \alpha_{io} - S_i^-, \qquad & i = 1, 2, \ldots, m \\
\sum_{j=1}^{n} \Lambda_j \beta_{rj} = t \beta_{ro} + S_r^+, \qquad & r = 1, 2, \ldots, s \\
\theta_0 = t - \frac{\sum_{i=1}^{m} \frac{S_i^-}{\alpha_{io}}}{m} \\
\Lambda_j \geq 0 \qquad & j = 1, 2, \ldots, n,
\end{aligned}
\tag{5}
$$

$$
\begin{aligned}
t &\geq 0, \\
S_i^- &\geq 0, \quad i = 1, 2, \ldots, m \\
S_r^+ &\geq 0, \quad r = 1, 2, \ldots, s
\end{aligned}
\tag{6}
$$

where θ_0 is the optimal value of the SBM model presented above(3).

$$
\begin{aligned}
& \max \beta_o = (\beta_{1o}, \beta_{2o}, \ldots, \beta_{so}) \\
& \text{subject to:} \\
& t + \frac{\sum_{j=1}^{s} \frac{S_r^+}{\beta_{ro}}}{s} = 1 \\
& \sum_{j=1}^{n} \Lambda_j \alpha_{ij} = t\alpha_{io} - S_i^-, && i = 1, 2, \ldots, m \\
& \sum_{j=1}^{n} \Lambda_j \beta_{rj} = t\beta_{ro} + S_r^+, && r = 1, 2, \ldots, s \\
& \theta_0 = t - \frac{\sum_{i=1}^{m} \frac{S_i^-}{\alpha_{io}}}{m} \\
& \Lambda_j \geq 0 && j = 1, 2, \ldots, n, \\
& t \geq 0, \\
& S_i^-, S_r^+ \geq 0, && i = 1, 2, \ldots, m, r = 1, 2, \ldots, s
\end{aligned}
\tag{7}
$$

Suppose that there is only one output parameter. We can derive a Parreto solution of the inverse SBM model presented by the following linear model:

$$
\begin{aligned}
& \max \beta_{1o} \\
& \text{subject to:} \\
& \sum_{j=1}^{n} \Lambda_j \alpha_{ij} = t\alpha_{io} - S_i^-, && i = 1, 2, \ldots, m \\
& \sum_{j=1}^{n} \Lambda_j \beta_{1j} = t\beta_{1o} + S_r^+, \\
& \theta_0 = 1 - \frac{\sum_{i=1}^{m} \frac{S_i^-}{\alpha_{io}}}{m} \\
& \Lambda_j \geq 0 && j = 1, 2, \ldots, n, \\
& S_i^- \geq 0, && i = 1, 2, \ldots, m
\end{aligned}
\tag{8}
$$

The equivalence between (5) and (6) for one output case is not trivial. We have provided a proof for the same in the later part of this document.

3.8 Stochastic Model

We realise that COVID-19 was an extremely dynamic scenario, and that the parameters changed quickly. This means that there doesn't really exist an all-time-best strategy to deal with the situation. It is wise to make modifications to our plan of action on the basis of the present environments. This motivates the use of Stochastic Modelling for the analysis. We could use immediate data to make edits in our decision making process and re valuate our decisions in real time using more data.

Here we calculate the stochastic efficiency using the symmetric error structure. First we introduce the chance constrained model related to input oriented stochastic CCR model for evaluating DMU_o:

$$\begin{aligned} &\min \theta \\ &\text{subject to:} \\ &p\{\sum_{j=1}^{n} \lambda_j \widetilde{x}_{ij} \leq \theta \widetilde{x}_{io}\} \geq 1-\alpha, \quad i = 1, 2, \ldots, m \\ &p\{\sum_{j=1}^{n} \lambda_j \widetilde{y}_{rj} \geq \widetilde{y}_{ro}\} \geq 1-\alpha, \qquad r = 1, 2, \ldots, s \\ &\lambda_j \geq 0 \qquad\qquad\qquad\qquad j = 1, 2, \ldots, n, \end{aligned} \tag{9}$$

Now we introduce symmetric error structure for random inputs and outputs:

Let

$$\begin{aligned} \widetilde{x}_{ij} &= x_{ij} + a_{ij}\widetilde{\epsilon}_{ij} \quad i = 1, 2, \ldots, m \\ \widetilde{y}_{rj} &= y_{rj} + b_{rj}\widetilde{\xi}_{rj} \quad r = 1, 2, \ldots, s \end{aligned} \tag{10}$$

Then we have:

$$\begin{aligned} \widetilde{x}_{ij} &\sim \mathcal{N}(x_{ij},\, \bar{\sigma}^2 a_{ij}^2) \\ \widetilde{y}_{rj} &\sim \mathcal{N}(y_{rj},\, \bar{\sigma}^2 b_{ij}^2) \end{aligned} \tag{11}$$

Using this we can convert the non-deterministic model presented above to a deterministic model.Using algebraic manipulation and normal distribution properties we convert non-linear deterministic model into a linear deterministic model:

$$
\begin{aligned}
&\min \theta^*(\alpha) = \theta \\
&\quad \text{subject to:} \\
&\quad \sum_{j=1}^{n} \lambda_j x_{ij} - \Phi^{-1}(\alpha)\bar{\sigma}(p_i^+ + p_i^-) \leq \theta x_{io}, i = 1, 2, \ldots, m \\
&\quad \sum_{j=1}^{n} \lambda_j y_{rj} + \Phi^{-1}(\alpha)\bar{\sigma}(q_r^+ + q_r^-) \geq y_{ro}, r = 1, 2, \ldots, s \\
&\sum_{j=1}^{n} \lambda_j a_{ij} - \theta a_{io} = p_i^+ - p_i^-, i = 1, 2, \ldots, m \\
&\sum_{j=1}^{n} \lambda_j b_{rj} - \theta b_{ro} = q_r^+ - q_r^-, r = 1, 2, \ldots, s \\
&\lambda_j, p_i^+, p_i^-, q_r^+, q_r^- \geq 0, j = 1, 2, \ldots, n, i = 1, 2, \ldots, m, r = 1, 2, \ldots, s
\end{aligned}
\tag{12}
$$

4 Results and Discussion

4.1 Analysing Results Obtained from DEA

State	BCC_Analysis	SBM_Analysis						
	Efficiency	Efficiency	s_doctors	s_nurses	s_hospitals	s_hospitalbeds	s_infections	s_recoveries
Andhra Pradesh	1	0.50367657	307.78252	618.4926	1.4252465	66.130055	0	0
Arunachal Pradesh	1.0013763	0.32010132	72.140239	411.95654	10.47169	109.91685	5.6847498	0
Assam	1.0070248	0.40499367	46.10558	52.43436	2.4896841	51.187632	0	0
Bihar	1.0093871	0.3683125	32.734459	19.599611	1.7712156	17.71733	0	0
Chhattisgarh	1	1	0	0	0	0	0	0
Delhi	1	1	0	0	0	0	0	0
Goa	1	1	0	0	0	0	0	0
Gujarat	1.0053789	0.37130857	198.58321	377.03072	5.5892921	67.881379	0	0
Haryana	1.0059035	0.6441038	14.846457	28.979164	0.46998369	9.3390017	0	0
Himachal Pradesh	1.0099724	0.3345673	39.259151	164.02563	5.1562543	102.43203	26.691412	0
Jammu & Kashmir	1	1	0	0	0	0	0	0
Jharkhand	1.0038815	0.52375273	4.8522864	6.9717143	10.774545	25.766997	0	0
Karnataka	1	0.33686464	405.20302	469.63405	73.264214	353.51337	85.394547	0
Madhya Pradesh	1.0041589	1	0	0	0	0	0	0
Maharashtra	1	0.46104854	149.95005	399.62027	0.67820245	95.671891	0	0
Mizoram	1	1	0	0	0	0	0	0
Nagaland	1	1	0	0	0	0	0	0
Odisha	1.0031392	0.38015496	0.99129389	5.4844971	1.3099466	18.304353	5.6527075	0
Punjab	1.020851	0.47166443	96.477268	216.66316	3.8881131	28.800462	18.617415	0
Rajasthan	1.0017793	0.35668901	60.827497	112.43187	1.1042915	28.025432	9.1740004	0
Sikkim	1.0052873	0.31635511	236.84136	914.68835	11.235001	238.78012	27.409235	0
Tamil Nadu	1.005112	0.43212718	11.929118	16.792793	0.2256894	20.794166	0	0
Kerala	1	0.47670161	220.63454	446.48116	2.6290854	42.507109	0	0
Uttar Pradesh	1.0076965	0.42441605	46.872815	117.96066	1.6967255	56.858928	4.6578406	0
Uttarakhand	1.0138096	0.37851669	7.4596289	15.261052	0.26489792	11.677459	1.7911458	0
West Bengal	1.0068185	0.37401057	81.627385	109.43699	5.0893865	62.901211	0	0
Tripura	1.0002132	0.47062714	52.646877	42.097452	2.0713659	27.952263	33.489076	0
Telengana	1.0014287	0.44700917	56.401132	52.034408	0.84187901	72.341102	0	0

Table 1: DEA-Efficiency-Analysis

The table provided displays the results obtained from running the BCC and SBM models on the collected dataset. In the BCC model analysis, it is evident that there is not a strict demarcation between the various states. Specifically, Andhra Pradesh, Chhattisgarh, Delhi, Goa, Jammu and Kashmir, Karnataka, Maharashtra, Mizoram, Nagaland, and Kerala all demonstrate an efficiency score of 1. The remaining states are not far behind either according to the BCC model's analysis.

Conversely, when applying the SBM model to the same dataset, a broader range of efficiency values emerges for the various states. Chhattisgarh, Delhi, Goa, Mizoram, and Nagaland also achieve an efficiency score of 1, while the lowest efficiency is approximately 0.31, observed in Sikkim, which was considered nearly efficient according to the BCC model.

Upon comparing the two models, it is apparent that the SBM model imposes somewhat stricter conditions for evaluating efficiencies. Notably, all states that are efficient under the SBM model are also efficient under the BCC model; however, the reverse is not necessarily true. This observation is empirically supported (for detailed information, please refer to the provided reference-go to: *insert reference*).

The SBM model is an extension of the BCC model, offering the advantage of assessing slack values associated with various parameters(as shown in the table). These slack values indicate the extent to which each parameter must be adjusted to achieve an efficiency score of 1. In our case, the model computes an optimal solution where the slack associated with output is zero. This outcome occurs because we have employed only one output parameter in the model. Under such conditions, the following lemma holds:

Lemma 4.1. *If an SBM model is applied to a set of Decision Making Units (DMUs) using multiple input parameters but only one output parameter, there is always an optimal solution where the slack associated with output is zero.*

Proof. Let P1 be the following problem(single output SBM DEA model):

$$\min \phi = t - \frac{\sum_{i=1}^{m} \frac{S_i^-}{x_{io}}}{m}$$

$$\begin{aligned} &\text{subject to:} \\ & t + \frac{s^+}{y_o} = 1 \\ & \sum_{j=1}^{n} \Lambda_j x_{ij} = t x_{io} - S_i^-, \qquad && i = 1, 2, \dots, m \\ & \Lambda Y = t y_o + s^+ \\ & \Lambda_j \geq 0 && j = 1, 2, \dots, n, \\ & t \geq 0, \\ & S_i^- \geq 0, && i = 1, 2, \dots, m \\ & s^+ \geq 0, \end{aligned} \tag{13}$$

We now have to show that: If ($\phi, S^-, s^+, \Lambda, t$) be an optimal solution to P1, then there exist $\hat{S^-}$ and $\hat{\Lambda}$ such that $(\phi, \hat{S^-}, 0, \hat{\Lambda}, 1)$ is a solution of P1.

Claim: $\hat{S^-} = S^- + s^+\frac{x_o}{y_o}$ and $\hat{\Lambda} = \Lambda$

Check for feasibility:We know that $x_0t = X\Lambda + S^-$, $y_0t = Y\Lambda - s^+$, $t = 1 - \frac{s^+}{y_0}$

$$\hat{S^-} = S^- + s^+\frac{x_o}{y_o} = x_0t - X\Lambda + (Y\Lambda - y_0t)\frac{x_0}{y_0}$$
$$\implies \hat{S^-} = Y\Lambda\frac{x_0}{y_0} - X\Lambda \ldots (1)$$

$$y_0t = Y\Lambda - s^+$$
$$\implies y_0(1 - \frac{s^+}{y_0}) = Y\Lambda - s^+$$
$$\implies y_0 = Y\Lambda \ldots (2)$$

From (1), (2) we see $(\phi, \hat{S^-}, 0, \hat{\Lambda}, 1)$ is a feasible solution of P1 using $\hat{S^-} = S^- + s^+\frac{x_o}{y_o}$ and $\hat{\Lambda} = \Lambda$.

Check for Optimality:

$$\text{Objective} = t - \frac{\sum_{i=1}^{m}\frac{S_i^-}{x_{io}}}{m} = 1 - \frac{\sum_{i=1}^{m}\frac{S_i^-}{x_{io}}}{m}$$
$$= 1 - \frac{\sum_{i=1}^{m}\frac{S_i^- + s^+\frac{x_0}{y_0}}{x_{io}}}{m}$$
$$= 1 - \frac{\sum_{i=1}^{m}(\frac{S_i^-}{x_{io}} + \frac{s^+}{y_0})}{m}$$
$$= 1 - t + \phi - \frac{s^+}{y_0} = \phi$$

Hence, we also see that $(\phi, \hat{S^-}, 0, \hat{\Lambda}, 1)$ is an optimal solution to P1. $\square$

Let P2 be the following problem:

$$\min \phi = 1 - \frac{\sum_{i=1}^{m}\frac{S_i^-}{x_{io}}}{m}$$

subject to:

$$\begin{aligned} &\Lambda X = x_o - S^-, \\ &\Lambda Y = y_o, \\ &\Lambda \geq 0, \\ &S^- \geq 0, \end{aligned} \tag{14}$$

Lemma 4.2. *If $\bar{\phi}$ is the optimal value of P2 then it is the optimal value of P1.*

Proof. Let $(\widetilde{S}_{-}, \widetilde{\Lambda})$ be the optimal value of P2.

So we have:

$$(\bar{\phi}, \widetilde{S}_{-}, \widetilde{\Lambda}) \rightarrow \text{optimal to P2.} \ldots (a)$$

$$\Rightarrow (\bar{\phi}, \widetilde{S}_{-}, 0, \widetilde{\Lambda}, 1) \rightarrow \text{feasible to P1} \ldots (b)$$

$$\exists(\widetilde{\phi}, \widetilde{S}_{-}, \widetilde{S}_{+}, \widetilde{\Lambda}, \widetilde{t}) \rightarrow \text{optimal to P1} \ldots (c)$$

$$\Rightarrow (\widetilde{\phi}, \widetilde{S}_{-} + \widetilde{S}_{+} * \frac{x_o}{y_o}, \widetilde{\Lambda}) \rightarrow \text{feasible to P2} \ldots (d)$$

From (a) and (d):

$$\widetilde{\phi} \geq \bar{\phi}$$

From (b) and (c):

$$\widetilde{\phi} \leq \bar{\phi}$$

Hence:

$$\widetilde{\phi} = \bar{\phi}$$

□

From the previous two lemmas, we conclude that solving P1 and P2 is equivalent for the single output DEA case in SBM.

It is also worth noting that all efficient states have associated slack values of zero, which aligns with expectations, as achieving an efficiency score of 1 has already been attained and there is no need to do anything for achieving the same.

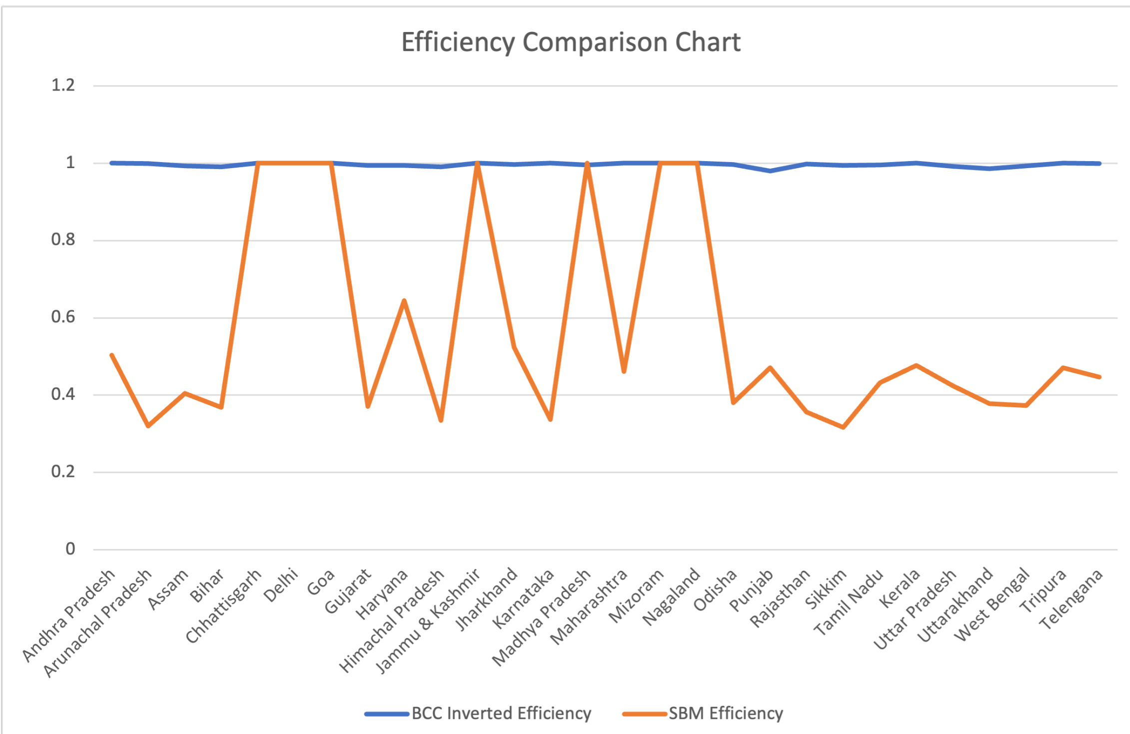

Figure 1: Plot of efficiencies obtained from BCC and SBM models

4.2 Analysing Results Obtained from InvDEA

	SBM_Analysis	BCC_Analysis
Andhra Pradesh	9.823525073	10.00000004
Arunachal Pradesh	10.00469709	9.999999715
Assam	10.00987331	9.999999935
Bihar	10.02397148	9.999999904
Chhattisgarh	3.771851349	10.00000006
Delhi	6.67304049	9.999999772
Goa	2.320150816	10.00000003
Gujarat	10.00290894	9.999999853
Haryana	10.00010712	10.00000036
Himachal Pradesh	10.00704992	9.999996937
Jammu & Kashmir	3.93027636	9.999999958
Jharkhand	10.03377316	9.999998436
Karnataka	9.803440641	10.00000145
Madhya Pradesh	10.00004515	10
Maharashtra	2.69008E-11	9.999999988
Mizoram	0.268023701	9.999999983
Nagaland	10.30991658	9.999998524
Odisha	9.928769299	9.999999656
Punjab	10.01287857	9.999999263
Rajasthan	9.918538913	9.99999924
Sikkim	10.0356175	9.999999887
Tamil Nadu	9.892639143	9.999999966
Kerala	3.470557848	9.999999997
Uttar Pradesh	10.00714704	10.00000001
Uttarakhand	10.00772573	10.00000007
West Bengal	10.00895139	10.00000023
Tripura	10.06014435	9.999999907
Telengana	10.00674691	9.999999838

Table 2: InvDEA-Analysis

The InvDEA SBM model is typically a non-linear model that requires iterative numerical procedures for solving. However, when the dataset contains only one output parameter, the model can be simplified. A lemma is presented as follows:

Lemma 4.3. *There exists a linear optimization model that is equivalent to the InvDEA SBM model when the dataset consists of a single output parameter.*

Proof. :A general inverse DEA model for non-radial DEA. □

The provided data table has been generated using the BCC and SBM InvDEA models. It offers a representative example by illustrating the necessary increase in the output parameter when all input parameters are augmented by 10%. It is observed that the SBM model predicts an approximate 10% increase in the output parameter in nearly every case to maintain the previously achieved efficiencies. In contrast, the BCC model predicts a significant variation in the output parameter.

Note that increasing all inputs by a constant amount is not mandatory and we have only provided a specific example. Different inputs can be increased by varying amounts and required increase in output can be determined to maintain efficiency levels. This will be done on the basis of input resource availability.

This data can serve as valuable guidance for making administrative decisions. It provides a basis for setting specific targets and objectives in alignment with the efficiency considerations obtained from these models.

5 Future Scope

In cases where suitable data can be collected, the use of stochastic models can be a valuable approach. The code has been developed for these models. Data availability however, has been a challenge. If appropriate data sets can be obtained or generated, running the stochastic model can provide insights about the effects of random variations and uncertainties.

The concept of an ”InvStochastic” model, which combines stochastic modeling with inverse optimization, could be explored. Implementing such a model would likely require estimations and informed assumptions for various parameters, as stochastic modeling often involves working with probabilistic distributions and simulating a range of possible outcomes.

Furthermore, feedback mechanisms based on the insights from stochastic models can be devised. These mechanisms can help in making dynamic and adaptive decisions, especially in scenarios where data is subject to change and uncertainty.

It’s also essential to account for noise in the data sets. Data quality is a critical factor in modeling and analysis. Given that not all data from open sources may be accurate, and some of it could potentially be manipulated, incorporating methods for data cleaning, validation, and error correction is necessary to ensure the reliability of the models and their outputs.

In summary, further work can be done to make the analysis proceed dynamically. Use of Stochastic models can help. Noise correction also needs to be implemented.

6 Appendix

6.1 SBM Code:

```
from pulp import *
import xlrd
import xlwt
from xlwt import Workbook
import numpy as np
m=int(input()) #This is the number of inputs we want
s=int(input()) #This is the number of outputs we want
n=int(input()) #This is the total number of DMUs
```

```
X=[]           #This is the matrix containing the inputs
Y=[]           #This is the matrix containing the outputs
loc = ("SURA_Final_Submit.xls")
wb = xlrd.open_workbook(loc)
sheet = wb.sheet_by_index(0)
wr=Workbook()
sheet1=wr.add_sheet('Sheet 1')
for i in range(n):
    sheet1.write(i+1,0,sheet.cell_value(i+1,0))
sheet1.write(0,1,'Efficiency')
for i in range(m):
    sheet1.write(0,i+2,"s_negative"+str(i))
for i in range(s):
    sheet1.write(0,i+m+2,"s_positive"+str(i))
for i in range(m):
    D=[]
    for j in range(n):
        D.append(sheet.cell_value(j+1,i+1))
    X.append(D)
for i in range(s):
    C=[]
    for j in range(n):
        C.append(sheet.cell_value(j+1,i+m+1))
    Y.append(C)
for k in range(n):
    o=k+1 #The dmu which you want to evaluate
    s_positive=[0]*s
    s_negative=[0]*m
    lamda=[0]*n
    prob=LpProblem("Efficiency Ranking",LpMinimize)
    for i in range(s):
        s_positive[i]=LpVariable("Slackpositive"+str(i),0)
    for i in range(m):
        s_negative[i]=LpVariable("Slacknegative"+str(i),0)
    for i in range(n):
```

```python
        lamda[i]=LpVariable("lamda"+str(i),0)
    t=LpVariable("Additive",0)
    s_positive=np.array(s_positive)
    s_negative=np.array(s_negative)
    lamda=np.array(lamda)
    X=np.array(X)
    Y=np.array(Y)
    dummy1=0
    for i in range(m):
        dummy1+=t*(1/m) - s_negative[i]*(1/(X[i][o-1]*m))
    prob+= dummy1 ,"objective"
    dummy2=0
    for i in range(s):
        dummy2+=t*(1/s) + s_positive*(1/(Y[i][o-1]*s))
    prob+= dummy2 == 1 , "constraint"
    lamda.transpose()
    lamdaX=np.matmul(X,lamda)
    lamdaY=np.matmul(Y,lamda)
    for i in range(m):
        prob+= lamdaX[i]== t*X[i][o-1] - s_negative[i],str(i)+"temp1"
    for i in range(s):
        prob+= lamdaY[i]== t*Y[i][o-1] + s_positive[i],str(i)+"temp2"
    prob.solve()
    sheet1.write(k+1,1,prob.objective.value())
    for i in range(m):
        sheet1.write(k+1,i+2,value(s_negative[i]))
    for i in range(s):
        sheet1.write(k+1,i+m+2,value(s_positive[i]))
wr.save('SBM_Final_Output.xls')
```

6.2 Inverse-SBM Code:

We compute the efficiency for the Inverse-SBM model using the SBM model discussed above. Then, after increasing input by a fixed amount, we utilize the Inverse-SBM to determine how much more output is needed to maintain a given level of efficiency.

```
#Inverse SBM DEA
sheet2= wb.sheet_by_index(1)
Z=[]
for i in range(m):
    D=[]
    for j in range(n):
        D.append(sheet2.cell_value(j+1,i+1))
    Z.append(D)
for k in range(n):
    o=k+1 #The dmu which you want to evaluate
    s_negative_i=[0]*m
    lamda_i=[0]*n
    prob_inverse=LpProblem("Increase in Output",LpMaximize)
    for i in range(m):  s_negative_i[i]=LpVariable("Slacknegative"+str(i),0)
    for i in range(n):
        lamda_i[i]=LpVariable("lamda"+str(i),0)
    beta=LpVariable("beta"+str(i),0)
    s_negative_i=np.array(s_negative_i)
    lamda_i=np.array(lamda_i)
    X=np.array(X)
    Y=np.array(Y)
    prob_inverse+= beta,"objective"
    dummy1=0
    for i in range(m):
        dummy1+=(1/m) - s_negative_i[i]*(1/((X[i][o-1]+Z[i][o-1])*m))
    prob_inverse+= dummy1==E[o-1],"constraint"
    lamda_i.transpose()
    lamdaX_i=np.matmul(X,lamda_i)
    lamdaY_i=np.matmul(Y,lamda_i)
    for i in range(m):
        prob_inverse+= lamdaX_i[i]== (X[i][o-1]+Z[i][o-1]) - s_negative_i[i],str(i)
   +"temp1"
    for i in range(s):
        prob_inverse+= lamdaY_i[i]== Y[i][o-1]+beta,str(i)+"temp2"
    prob_inverse.solve()
```

```
    sh2.write(k+1,1,(value(beta)/Y[i][o-1])*100)
wr.save('Output_Increase(Percent).xls')
```

6.3 BCC Code:

```
from pulp import *
import xlrd
import xlwt
from xlwt import Workbook
import numpy as np
m=int(input()) #This is the number of inputs we want
s=int(input()) #This is the number of outputs we want
n=int(input()) #This is the total number of DMUs
X=[]           #This is the matrix containing the inputs
Y=[]           #This is the matrix containing the outputs
loc = ("BCC_DEA.xls")
wb = xlrd.open_workbook(loc)
sheet = wb.sheet_by_index(0)
wr=Workbook()
sheet1=wr.add_sheet('Sheet 1')
for i in range(n):
    sheet1.write(i+1,0,sheet.cell_value(i+1,0))
sheet1.write(0,1,'Efficiency')
for i in range(m):
    D=[]
    for j in range(n):
        D.append(sheet.cell_value(j+1,i+1))
    X.append(D)
for i in range(s):
    C=[]
    for j in range(n):
        C.append(sheet.cell_value(j+1,i+m+1))
    Y.append(C)
for k in range(n):
    o=k+1 #The dmu which you want to evaluate
```

```
    lamda=[0]*n
    prob=LpProblem("Efficiency Ranking",LpMaximize)
    for i in range(n):
        lamda[i]=LpVariable("lamda"+str(i),0)
    t=LpVariable("Additive",0)
    lamda=np.array(lamda)
    X=np.array(X)
    Y=np.array(Y)
    prob+= t ,"objective"
    lamda.transpose()
    lamdaX=np.matmul(X,lamda)
    lamdaY=np.matmul(Y,lamda)
    for i in range(m):
        prob+= lamdaX[i]<= X[i][o-1],str(i)+"temp1"
    for i in range(s):
        prob+= lamdaY[i]>= t*Y[i][o-1],str(i)+"temp2"
    dummy1=0
    for i in range(n):
        dummy1+=lamda[i]
    prob+= dummy1 == 1 , "constraint"
    prob.solve()
    sheet1.write(k+1,1,prob.objective.value())
wr.save('BCC_Efficiency.xls')
```

6.4 Inverse-BCC Code:

We compute the efficiency for the Inverse-BCC model using the BCC model discussed above. Then, after increasing input by a fixed amount, we utilize the Inverse-BCC to determine how much more output is needed to maintain a given level of efficiency.

```
#Inverse DEA code
sheet2= wb.sheet_by_index(1)
Z=[]
for i in range(m):
    D=[]
```

```
    for j in range(n):
        D.append(sheet2.cell_value(j+1,i+1))
    Z.append(D)
for i in range(s):
    sheet1.write(0,i+2,"beta"+str(i))
for k in range(n):
    o=k+1
    weights=[]
    for i in range(s):
        weights.append(1/s)
    lamda=[0]*n
    beta=[0]*s #Here beta stands for increment in output for a particular increment
     in input
    prob_inverse=LpProblem("Max increase in Outputs",LpMaximize)
    for i in range(n):
        lamda[i]=LpVariable("lamda"+str(i),0) #All lamda are greater than 0
    for i in range(s):
        beta[i]=LpVariable("beta"+str(i),0) #All beta are greater than 0
    dummy1=0
    for i in range(n):
        dummy1+=lamda[i]
    prob_inverse+= dummy1 == 1 , "constraint" #Summation of lamda is equal to 1
    dummy2=0
    for i in range(s):
        dummy2+=beta[i]*weights[i]
    prob_inverse+= dummy2 ,"objective"
    lamda=np.array(lamda)
    lamda.transpose()
    lamdaX=np.matmul(X,lamda)
    lamdaY=np.matmul(Y,lamda)
    for i in range(m):
        prob_inverse+= lamdaX[i]<= X[i][o-1]+Z[i][o-1],str(i)+"temp1"
    for i in range(s):
        prob_inverse+= lamdaY[i]>= (Y[i][o-1]+beta[i])*E[o-1],str(i)+"temp2"
    prob_inverse.solve()
```

```
    for i in range(s):
        sheet1.write(k+1,i+2,(value(beta[i])/Y[i][o-1])*100)
wr.save('Output_increase_BCC.xls')
```

6.5 Stochastic Code:

```
from pulp import * import xlrd import xlwt
from xlwt import Workbook
import numpy as np
m=int(input())          #This is the number of inputs we want
s=int(input())          #This is the number of outputs we want
n=int(input())          #This is the total number of DMUs
phi=float(input())      #This represents the confidence level
sigma=float(input())    #This represents the variance
X=[]                    #This is the matrix containing the mean of inputs
Y=[]                    #This is the matrix containing the mean of outputs
A=[]                    #This is the matrix containing the variance factor of inputs
B=[]                    #This is the matrix containing the variance factor of inputs
loc = ("Stochastic_DEA.xls")
wb = xlrd.open_workbook(loc)
sheet = wb.sheet_by_index(0)
sheet_var = wb.sheet_by_index(1)
wr=Workbook()
sheet1=wr.add_sheet('Sheet 1')
for i in range(n):
    sheet1.write(i+1,0,sheet.cell_value(i+1,0))
sheet1.write(0,1,'Efficiency')
for i in range(m):
    D=[]
    for j in range(n):
        D.append(sheet.cell_value(j+1,i+1))
    X.append(D)
for i in range(m):
    D=[]
    for j in range(n):
```

```
        D.append(sheet_var.cell_value(j+1,i+1))
    A.append(D)
for i in range(s):
    C=[]
    for j in range(n):
        C.append(sheet.cell_value(j+1,i+m+1))
    Y.append(C)
for i in range(s):
    C=[]
    for j in range(n):
        C.append(sheet_var.cell_value(j+1,i+m+1))
    B.append(C)
for k in range(n):
    o=k+1 #The dmu which you want to evaluate
    p_positive=[0]*m
    p_negative=[0]*m
    q_positive=[0]*m
    q_negative=[0]*m
    lamda=[0]*n
    prob=LpProblem("Efficiency Ranking",LpMinimize)
    for i in range(s):
        q_positive[i]=LpVariable("q_positive"+str(i),0)
    for i in range(s):
        q_negative[i]=LpVariable("q_negative"+str(i),0)
    for i in range(m):
        p_postive[i]=LpVariable("p_positive"+str(i),0)
    for i in range(m):
        p_negative[i]=LpVariable("p_negative"+str(i),0)
    for i in range(n):
        lamda[i]=LpVariable("lamda"+str(i),0)
    p_positive=np.array(p_positive)
    p_negative=np.array(p_negative)
    q_positive=np.array(q_positive)
    q_negative=np.array(q_negative)
    lamda=np.array(lamda)
```

```
    X=np.array(X)
    Y=np.array(Y)
    A=np.array(A)
    B=np.array(B)
    theta=LpVariable("Stochastic",0)
    prob+= theta ,"objective"
    lamda.transpose()
    lamdaX=np.matmul(X,lamda)
    lamdaY=np.matmul(Y,lamda)
    lamdaA=np.matmul(A,lamda)
    lamdaB=np.matmul(B,lamda)
    for i in range(m):
        prob+= lamdaX[i] - phi*sigma*(p_positive[i]+p_negative[i])<= theta*X[i][o
-1],str(i)+"Mean_x"
    for i in range(m):
        prob+= lamdaA[i] - theta*A[i][o-1]== (p_positive[i]-p_negative[i]),str(i)+"
Var_x"
    for i in range(s):
        prob+= lamdaY[i] + phi*sigma*(q_positive[i]+q_negative[i])>= Y[i][o-1],str(
i)+"Mean_y"
    for i in range(s):
        prob+= lamdaB[i] - B[i][o-1]== (q_positive[i]-q_negative[i]),str(i)+"Var_y"
    prob.solve()
    sheet1.write(k+1,1,prob.objective.value())
    wr.save('Stochastic_Efficiency.xls')
```

7 References

1. **Data Envelopment Analysis**

   - *Kaoru Tone* A slacks-based measure of efficiency in data envelopment analysis(European Journal of Operational Research 130 (2001) 498-509)
   - *Kshitish Kumar Mohanta, Deena Sunil Sharanappa, Abha Aggarwal* Efficiency analysis in the management of COVID-19 pandemic in India based on data envelopment analysis(Current Research in Behavioral Sciences 2 (2021) 100063)
   - *Nahia Mourada, Ahmed Mohamed Habibb and Assem Tharwata* Appraising healthcare systems' efficiency in facing COVID-19 through data envelopment analysis(Decision Science Letters 10 (2021) 301–310)

2. **Inverse Data Envelopment Analysis**

   - *Sabri Boubaker, Tu D.Q. Le and Thanh Ngo* Managing bank performance under COVID-19: A novel inverse DEA efficiency approach(International Transactions in Operational Research (2022) 1–17)
   - *GuoJun Zhang, JinChuan Cui* A general inverse DEA model for non-radial DEA(Computers & Industrial Engineering 142 (2020) 106368)

3. **Stochastic Data Envelopment Analysis**

   - *Susan X. Li* Stochastic models and variable returns to scales in data envelopment analysis(European Journal of Operational Research 104 (1998) 532-548)
   - *Behzadi, M.H. , Mirbolouki, Mahnaz* Symmetric Error Structure in Stochastic DEA(Int. J. Industrial Mathematics 4 (2012) 335-343)

4. **Data Sources Used**

   - National Health Profile 2022 - Ministry of Health, GOI
   - Population Density Data, Census 2011
   - Covid-19 Data Regarding Infections, Deaths, Recoveries and Vaccinations in India, GOI