For office use only	Team Control Number	For office use only
T1 ________________	**87207**	F1 ________________
T2 ________________		F2 ________________
T3 ________________	Problem Chosen	F3 ________________
T4 ________________	**E**	F4 ________________

2018
MCM/ICM
Summary Sheet

Summary

The center of our work is to model the interaction between human and environmental factors and their joint contribution to a state's fragility. In order to guarantee the interpretability and verifiability of our model, we specifically formulated a concrete theoretical model framework that is flexible and applicable to many stiuations. We then implemented the framework in details.

We proposed a probabilistic model framework in Section 2.3 for scoring a country's fragility, combining current well-recognized methods to identify a country's extent of political and environmental fragility. Our fragility score is consistent with traditional methods, and yields more comprehensive information about a country's fragility.

We were then able to dive into the complex dynamics between factors. In Section 2.4, we used theoretically well supported and verifiable tools, e.g. PSM, to measure the impact of direct environmental factors on a state's fragility. We proposed and verified two hypothesis models for indirect effects: the mediator variable model and the moderator variable model. A case study of Iraq gives us insight in how the dynamics take place.

We further developed a temporal model in Section 2.5 to describe the dynamics of environmental change with the presence of human intervention. Specifically, we use ARIMA, a widely used time series model to forecast a country's GDP growth, and based on this, we modeled climate change based on its autoregressive property and its direct and indirect relation between GDP.

Our model produced fruitful and insightful results. For example, we found the different patterns of impact of climate change for fragile and stable states in Section 3.3, and derived the interaction between envrionment and multiple human factors consistent with interdisciplinary knowledge. The results are informative for policy making process. In Section 3.5, we discussed the trade-off rapid development and environmental protection, and discovered that mediocre economic growth can balance environmental deterioration in the long term.

Balancing Fragility in the Long Run

Tai Wang, Hao Zhou, Rui Feng
Zhejiang University

1 Introduction

Climate change has become a common concern for a large portion of the human community. As such, analyzing the effect of climate change and its relation to a state's fragility and people's welfare drew many attentions from researchers.

Quantifying the fragility of states based on human factors, such as the FSI score has been extensively studied by researchers and instutitions, such as in [1, 2]. However, these models do not consider the impact of environmental factors. For example, deterioation in natural environment may contribute on regional instability and violence [3, 4, 5]. As environmental factors are important in determining a state or a region's sustainability, merely considering human factors is clearly insufficient.

Our work combined previous efforts to incorporate human factors and environmental factors into a novel fragility score that is consistent with traditional results. We also analyzed the effects of environmental factors, both indirect and direct. We also forecasted the future climate change of an example country, and found that moderate economic development balances environmental damage in the long run. Our results are insightful for policy-makers.

First we propose a theoretical framework of our model in Section 2. Then, implementation of our framework, experiment designs and results are thoroughly listed and discussed in detail in Section 3. We discuss the strength and weakness of our model, parameter sensitivity, and the relation of our work to interdiscriplinary works in the field of environmental science in Section 4.

2 Theoretical Framework

In this part we propose theoretical framework for our analysis of the impact of climate change on *state fragility*.

2.1 Assumptions and Model Framework

Our hypothesis framework is illustrated in Figure 1. We propose two natural assumptions, based on which we derive the basic framework of our model.

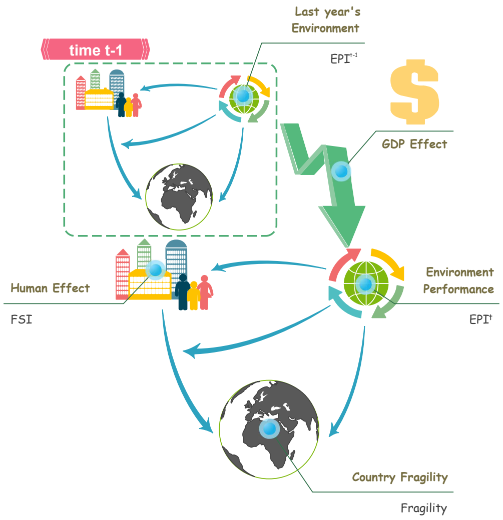
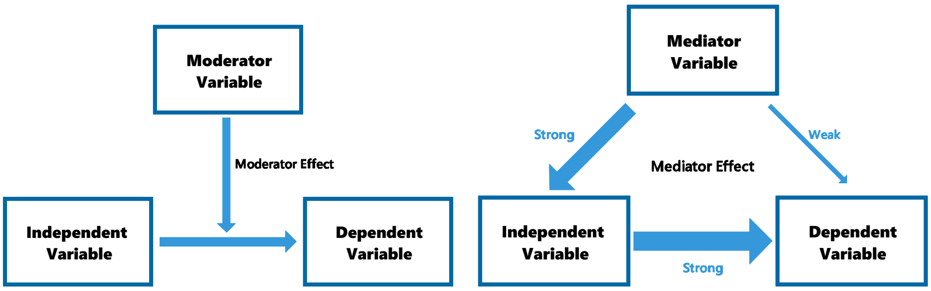


Figure 1: Hypothesis Model Illustration



(a) Moderator variable model

(b) Mediator variable model

Figure 2: Hypothesis models for indirect effect of environmental factors. E represents environmental factors, H represents human factors, and F represents state fragility. The fragility is jointly determined by E and H, by two hypothesis approach.

1. The *state fragility*, a concept to estimate the sustainability of states, is dependent on and only on *human factors* and *environmental factors.*
2. The environmental factors and human factors interact with each other.

The assumptions are natural. Assumption 1 requires to quantify the fragility which considers both human factors and environmental factors. We propose a novel framework to quantify fragility by incorporating the human and environmental factors into a probabilistic framework.

Assumption 2 requires a more sophisticated analysis of the two factors, including their respective and joint effects on the fragility, and the interaction between them. We are especially interested in the effects of environmental factors, which include *direct* effect, which is the influence on fragility directly imposed by environmental factors; and *indirect* effect, which is the influence on fragility imposed by envirionmental factors indirectly through human factors. Two hypothesis model to explain the indirect effect is visualized in Figure 2(a) and 2(b).

The direct effect of environmental factors is measured by the effect of environmental factors on fragility score, with an unbiased estimation of the effect obtained by propensity score matching [6, 7, 8]. The indirect effect of environmental factors can be explained by two hypothesis models: the *moderator variable* model, as shown in Figure 2(a); and the *mediator variable* model, as shown in Figure 2(b). We verify these two hypothesis.

In the following of this section, we are dedicated to enrich our model by developing several key ingredients:

1. A novel fragility score measure incorporating both environmental and human factors;
2. The interaction pattern between human factors, envirionmental factors, and the fragility;
3. The temporal model of a state's environmental status.

The basic framework is sufficient to cover most of the requirements of the tasks.

2.2 Representing the Two Factors

Notation	Description
H	random variable of human factors
E	random variable of environmental factors
F	binary random variable of fragility
s_f	fragility score

Table 1: Notations

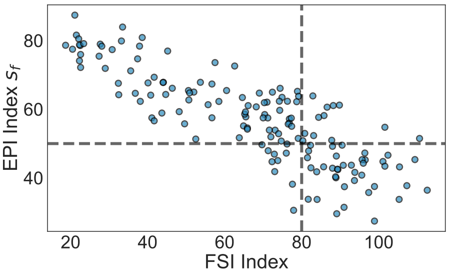


Figure 3: EPI and FSI Indexes

Basic notations are listed in Table 1. The variables in the table are theoretical. In order to reflect these variables numerically from empirical data, we introduce two widely recognized indexes:

Environmental Performance Index (EPI). It is an index to evaluate a state's environmental performance developed by Yale University [2]. It is composed of indicators in ecosystem vitality and environmental health. The higher the index, the better the environment,

Fragile States Index (FSI). It is an index to measure a country's vulnerability to conflict developed by Fund for Peace and Foreign Policy by accounting for 12 social, economical, and political indicators. [1] Higher index indicates more vulnerability.

FSI index uses indicators relevant to human activity, and is therefore used as empirical data for H. EPI index measures environmental fragility, and is therefore used as empirical data for E.

Sometimes, we use binarized values of EPI and FSI indexes to denote the status of a state. The threshold and the relation between EPI and FSI are visualized in Figure 3.

2.3 Probabilistic Fragility Measure

In this part, we derive a novel fragile score, $\mathrm{s_f}$, which incorporates both environmental and human factors. The score $\mathrm{s_f}$ is based on probabilistic intuitions, and is therefore called the *probabilistic fragility score*, or the fragility score for convenience.

Without loss of generality, we refer to regions, sovereign states, and other concerned geographical entities as states.

We assume that a state either fragile or stable, described by a binary random variable F, where $\mathrm{F} = 1$ if the considered state fragile, and $\mathrm{F} = 0$ if it is stable. H and E are random variables describing the human and environmental factors of the state. For convenience, we further assume that E is binary, i.e. $\mathrm{E} = 1$ when the state's environment is sustainable, and $\mathrm{E} = 0$ when it is not.

Consider the probability of a state being fragile, given its human and environmental factors:

$$\mathbb{P}(\mathrm{F} = 1|\, \mathrm{E} = e, \mathrm{H}) \;\; (e = 0, 1) \tag{1}$$

The probability given in 1 quantifies the extent of fragility of the state, given certain environmental and human factors. It is higher when the state is more vulnerable. However, the conditional distribution is hard to estimate. We then factorize it into a more easily calculated form:

$$\mathbb{P}(\mathrm{F} = 1|\, \mathrm{E} = e, \mathrm{H}) = \frac{\mathbb{P}(\mathrm{E} = e, \mathrm{F} = 1|\, \mathrm{H})}{\mathbb{P}(\mathrm{E} = e|\, \mathrm{H})} = \frac{\mathbb{P}(\mathbf{Z} = 1|\, \mathrm{H})}{\mathbb{P}(\mathrm{E} = e|\, \mathrm{H})} \;\; (e = 0, 1) \tag{2}$$

In which we defined a new random variable $\mathbf{Z} = 1$ if $\mathrm{E} = e, \mathrm{F} = 1$ and $\mathbf{Z} = 0$ otherwise. Eqn. 2 allows us to only estimate the conditional probability of two binary random variables given human factors H.

We assume a linear assumption and perform a logistic regression to estimate the probabilities:

$$
\begin{aligned}
\hat{p}_z &= \frac{\exp(\mathbf{W}_1\,\mathrm{H})}{1+\exp(\mathbf{W}_1\,\mathrm{H})} \\
\hat{p}_e &= \frac{\exp(\mathbf{W}_2\,\mathrm{H})}{1+\exp(\mathbf{W}_2\,\mathrm{H})}
\end{aligned} \tag{3}
$$

In order to make the estimated probability distribution resemble the true distribution, we estimate parameters $\mathbf{W}_1, \mathbf{W}_2$ by minimizing the cross entropy loss, which is equivalent to minimizing the KL divergence [9] between the estimated distribution and the empirical distribution. Specifics of optimization are omitted; interested readers may see [9].

Finally, probabilities in Eqn. 2 is replaced by the estimates given in Eqn. 3, yielding the *probabilistic fragility score*:

$$
\mathrm{s_f} = \frac{\hat{p}_z}{\hat{p}_e} \tag{4}
$$

Notes on the fragility score. The fragility score, $\mathrm{s_f}$, is derived based on probabilistic intuitions and linear assumptions. Higher $\mathrm{s_f}$ indicates higher risks of being fragile. However, the score $\mathrm{s_f}$ can be larger than one, and is, therefore, not in form of probability. However, it does not hurt its applicability: if the estimated $\hat{p}_z$ is larger than $\hat{p}_e$, we have even more reasons to believe that the considered state is fragile.

2.4 Modeling the Interaction between Variables

We then begin to model the relationship between the three variables: environmental factors E, human factors H, and the fragility score $\mathrm{s_f}$.

Direct Effect of Environmental Factors

In order to measure the effect of environmental factors on fragility score, a naive approach would be to sample states of both sustainable environment and unsustainable environment, and compare their average fragility score. Formally, write s_f^0 as the average fragility score of the sustainable group, and s_f^1 as the average fragility score of the unsustainable group. One then compares the difference $s_f^* = s_f^0 - s_f^1$.

The above approach gives, however, biased estimation, because the apparent difference between these two groups may be depend on human factors that affected whether or not a state's environment is sustainable, instead of the environmental status per se. For example, the approach might compare scores of the states that are environmentally unsustainable and in political turmoil, with the scores of the states that are environmentally sustainably and politically stable. The difference of political status results in unbiased estimate of the effect of environmental factors.

To control for the differences of human factors between the sustainable group and the unsustainable group, we use the propensity score matching, a statistical technique that attempts to estimate unbiasly the effect of a variable, in this case, the environmental status.

Formally, the propensity score of a certain state is defined as the conditional probability of the environmental status given its human factors,

$$p = \mathbb{P}(\mathrm{E} = 1 | \, \mathrm{H}) \tag{5}$$

In order to estimate the probability, we adopt similar procedures used in Section 2.3, using the score of logistic regression, $\hat{p}$, of human factors H against environmental status E. Then we match each of the unsustainable states to one sustainable state on propensity score, by using *Nearest Neighbor Matching*: each unsustainable state is matched to the sustainable state whose propensity score is the closest. As such, a new data set in which the sustainable group and the unsustainable group and their propensity scores are balanced, is obtained.

Based on the newly obtained data set, we calculate the adjusted score difference:

$$\hat{s}^*_f = \hat{s}^0_f - \hat{s}^1_f. \tag{6}$$

Where $\hat{s}^0_f$, $\hat{s}^1_f$ are average scores of the sustainable and unsustainable group, drawn from the data set obtained by PSM.

Indirect Effect of Environmental Factors

In order to measure the indirect effect of environmental factors E, we propose two candidate models: the moderator variable model, and the mediator variable model.

The moderator variable model assumes that the environmental factors influence the fragility score, and the relationship is calibrated by the effect of human factors. In this case, the human factors H are called *the moderator variable*, as illustrated in Figure 2(a). [10]

The mediator variable model assumes that the environmental factors and human factors jointly influence the fragility score. Furthermore, the environmental factors influence the fragility score both directly, and indirectly through the human factors. The illustration of this model is as in Figure 2(b). [11]

Moderator variable model. The model is written as

$$\mathrm{s_f} = \mathbf{W}_1 \,\mathrm{H} + W_2 \,\mathrm{E} + \sum_{i=1}^{p} \beta_i h_i \,\mathrm{E} \tag{7}$$

where $\mathrm{H} = [h_1, \ldots, h_p]'$, and h_i is the ith component of H, indicating a specific factor. $\mathbf{W}$ and β_i are parameters.

The term, $\sum_{i=1}^{p} \beta_i h_i$, represents the effect of each human factor on the relation between the environmental factor and the fragility score, since by taking partial derivative, we observe that

$$\frac{\partial s_f}{\partial \mathrm{E}} = W_2 + \sum_{i=1}^{p} \beta_i h_i$$

The derivative indicates that the effect of environmental factor E comes from both itself, described by W_2, and each human factor h_i, described by β_i. If the factor h_i has no effect on the relation, then β_i should be close to zero. Consider, therefore, the following statistical test:

$$H_0 : \beta_1 = \beta_2 = \ldots = \beta_p \tag{8}$$

The rejection of H_0 shows that the moderator effect of H is significant. Furthermore, we wish to specifically investigate the effect of each factor. Hence in the following test,

$$H_0 : \beta_i = 0 \tag{9}$$

if H_0 is rejected, we are confidence to say that the moderator effect of h_i is statistically significant.

If the moderator effect is indeed significant, then the extent of the effect of H is then quantified by $\beta = [\beta_1, \ldots, \beta_p]'$ and respective components.

Mediator variable model.

We first declare the following values:

- b_1: the coefficient of the linear regression model, in which E predicts $\mathrm{s_f}$.
- b_2: the coefficient of the linear regression model, in which H predicts $\mathrm{s_f}$.
- b_3: the coefficient of E in the linear regression model in which E and H jointly predicts $\mathrm{s_f}$.
- b_4: the coefficient of H in the linear regression model in which E and H jointly predicts $\mathrm{s_f}$.

First, we need to verify by statistical tests that b_1 and b_2 are significantly nonzero.

If H indeed acts as a mediator variable, by model assumption, b_4 is significantly nonzero. Furthermore, since E indirectly influences $\mathrm{s_f}$ through H, the explaining power of E alone should be reduced once H is introduced. In this case, $b_3 < b_1$.

If all the above four conditions hold significantly, we are confident to say that human factors act as mediator variables, through which the environmental factors influence the fragility score.

2.5 Temporal Model

In order to better model climate change, we further consider time as a variable, and investigate how climate change would have influenced the fragility of states. We begin by formulating the following assumptions:

- The evolution of climate change is a Markov process, i.e. E_t is dependent on E_{t-1} and independent of $k < t-1$.
- Human factors act as a moderator of EPI: it moderates the relation between E_{t-1} and E_t.
- Human factors are majorly embodied in economic status.

The first hypothesis is for convenience of modeling. The term "moderator" in the second hypothesis is in consistence with the definition in Section 2.4. The third hypothesis is because we consider factors at the country level, therefore we reasonably assume that economic status is representative.

The idea of the hypothesis is simple: one may imagine that today's environment depends on yesterday's environment. Without invervention, the environment evolves all by itself. With the presence of human activity, the evolution of environment is "moderated" by human factors.

We use EPI index as an indicator of environmental status and establish a temporal model of EPI to investigate climate change. Specifically, we choose Mauritius as an example state, and illustrate how and when it would reach a tipping point. The basic idea of our model is as illustrated in the formula below:

$$E_t = \beta_0 E_{t-1} + \beta_1 H_t + \beta_2 H_t \times E_{t-1} \tag{10}$$

In which we use the subscript t to denote the value at time t. The first term embodies the autoregressive property of the environmental factor: its present status depends on its previous status. THe second term formulates human factors. Since we assumed that economic status is representative, in experiments in Section 3.5, we would use GDP and GDP growth as indicators of human factors. The third term represents the mediator effect of human factors, derived from the second hypothesis.

One may notice that predicting E_t requires knowing H_t a priori, which is impossible. In order to approximate the evolution of climate change, we establish another temporal model to predict H_t, which is, in this case, GDP growth.

Modeling GDP growth We propose using ARIMA, a typical model widely used for time series forcasting, to model the time series of GDP growth rate. Generally, the model establishes that

$$Y_t = \sum_{i=1}^{p} a_i Y_{t-i} + \sum_{i=0}^{q} b_i \epsilon_{t-i} \tag{11}$$

$$Y_t = E_t - E_{t-k} \tag{12}$$

where p, q, k are hyperparameters, a_i, b_i are parameters to be estimated. The first sum in Eqn 11 is the autoregressive term, and the second sum is the moving average of white noises, where $\epsilon_{t-i} \sim \mathrm{N}(0, 1)$. Eqn. 12 performs differencing, where k is the order of differencing, to guarantee that the time series to be modeled is stationary.

Due to lack of space, the detailed description of this model is omitted. We encourage inerested readers to see detailed description in [12].

Modeling Government Invervention and Economic Boom Where the environment deteriorates, the government should step in to prevent the environment from becoming fragile. History of the development of many countries, such as UK and China, shows that fast economic development could be at the expense of environment performance. This is in consistence with the experiment results of our model, discussed in detail in Section 3.5.

The forcase of GDP growth is generally stable by our model, as shown in Figure 10(a). To model fast economic development at the cost of environment, we introduce the following two parameters:

- α: the government investment to neutralize environment deterioration;
- μ: the economic boom factor.

Specifically, the economic boom brings additional μ GDP growth every year. The government uses α of the annual GDP to ease the negative impact of fast economic development. It does not mean that the annual GDP or GDP growth rate is decreased by α; the effect of government intervention is manifested in EPI. The specific implementation is described in Section 3.5.

In reality, the growth rate of GDP doesn't necessarily increase at a regular speed. In settings of fast economic development, for example, China, the GDP growth rate is jumped to a high level which is sustained for quite a long period of time, instead of growing steadily to a high level. However, our model setting is sufficient for stimulating the effect of high economic growth.

3 Experiments

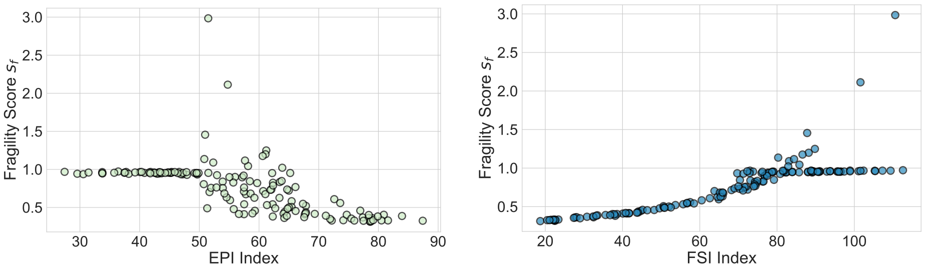
3.1 Data Preparation

Dataset. The data set we prepared include the following indicators: FSI Index, EPI Index, Gross Domestic Product (GDP) (constant 2010 US dollar), and GDP growth rate. The indexes are acquired from [1] and [2]. GDP and GDP growth rate are acquired from World Bank Databank [13]. For each of them, we prepared data for 149 countries from 2007 to 2017.

3.2 Calculation of Fragility Score

In order to calculate fragility score defined in Section 2.3, we need to identify, a priori, which states are fragile and which states are environmentally unstable. In order to achieve so, we determine that a state is fragile, i.e. $\mathrm{F} = 1$, if its FSI score is higher than a certain threshold F_0, and a state is environmentally fragile, i.e. $\mathrm{E} = 1$, if its EPI score is lower than a certain threshold E_0. The thresholds are chosen differently for each year, because the indexes of different years aren't necessarily calculated using the same methodology. Therefore, thresholds for each year are chosen to guarantee that the fragile states and environmentally fragile states occupy approximately 30% of the states, respectively. As such, each state's status of fragility and environmental fragility is approximated by FSI and EPI indexes. Furthermore, we use the 12 indicators used in the calculation of FSI, specified in Section3.1 as components of human factors $\mathrm{H} = [h_1 \ldots h_1 2]$.

Hence, all variables needed for the calculation of our fragility score, including $\mathrm{H}, \mathrm{E}, \mathrm{F}$, are specified for each state. Logistic regression was run as described in Section 2.3 to obtain fragility score of each state. Finally, we visualize the relationship between the score and the two indexes in Figure 4. Figure 4(a) shows that states with lower EPI indexes have higher scores in general, but the variance is high when the environment is worsened. Figure 4(b) shows that states with higher FSI obtain higher scores. The figures show that FSI majorly determines $\mathrm{s_f}$, while EPI acts as an adjustment.



(a) EPI (b) FSI

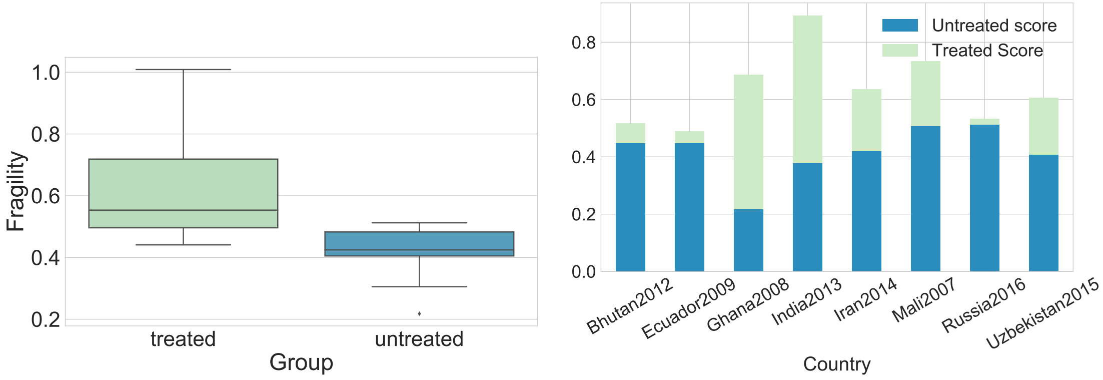
Figure 4: Relationship between Fragility Score $\mathrm{s_f}$ and FSI and EPI.

Consistency Test. In order to show that $\mathrm{s_f}$ is a reasonable criterion of states' fragility, we need to make sure that a state which is completely better than another state, i.e. with higher EPI and lower FSI, obtains lower $\mathrm{s_f}$. In this case, we call that the two states are *inconsistent*. We therefore define the average reverse number order r:

$$r \triangleq \frac{1}{2N}\sum_{k=1}^{N}\frac{r_k}{N} = \frac{1}{2N^2}\sum_{k=1}^{N} r_k \tag{13}$$

where for the kth state in the dataset, r_k is the number of other states that inconsistent with it.

The r_k calculated for the indexes in the year of 2017 is 0.01764, which is sufficiently low to bring confidence to the fragility score s_f.



(a) General Effect (b) Specific Countries

Figure 5: Direct Effect of Environmental Factors.

3.3 Indirect Effect of Environmental Factors

By observing the relation between EPI index and fragility in Figure 4(a), we find that environmental factors exhibit different kinds of influence at different stages. When EPI is small, the change of fragility with respect to EPI is little. When EPI is higher than a certain threshold, $E_0 = 50$, the changes of EPI begins to induce changes of fragility score. It could be explained, since when a state's environmental performance is too low, it tends to become too unstable such that further deterioration would have minor effect.

Therefore, we establish different models for states which are below and higher than the threshold, respectively, to explain the indirect effect of environmental factors in terms of moderator and mediator effects formulated in Section 2.4.

When the Index $\mathrm{EPI} > 50$

These states are comparably environmentally stable. For these states, change of environment significantly influence fragility. The relationship is illustrated in Figure 6.

Moderator Effect To examine the moderator effect, we establish a linear regression model:

$$\mathrm{s_f} \sim \mathrm{EPI} + \mathrm{F} \times \mathrm{EPI} + \mathrm{F} \tag{14}$$

where

$$S = \sum \mathrm{E_k} + \sum \mathrm{P_k} + \sum \mathrm{S_k} + \sum \mathrm{C_k} + \sum \mathrm{X_k} \tag{15}$$

is the sum of economic, political, social, cohesion indicators, and external indicators used in FSI index, as in Table [1]. The second term is the effect of moderators.

We select significant variables from the regression model in Eqn. 14 by bidirectional step regression. The selected variables, their coefficients, p-value and level of significance are listed in Table 2.

Var.	Coef.	P-val.	Level
EPI	0.050534	7.44e-11	Super
E1	0.337434	0.003849	High
E2	-0.404308	0.002120	High
P3	0.363195	0.009289	High
EPI×E1	-0.004889	0.006607	High
EPI×E2	0.006458	0.002109	High
EPI×P3	-0.005600	0.011503	Medium
EPI×S2	-0.004610	0.000142	High

Table 2: Moderator Effect when $\mathrm{E} > 50$

We see that E1, E2, P3 appears both individually in Table 2 and also as factors of cross products. Therefore, EPI moderates the relations between these variables and fragility.

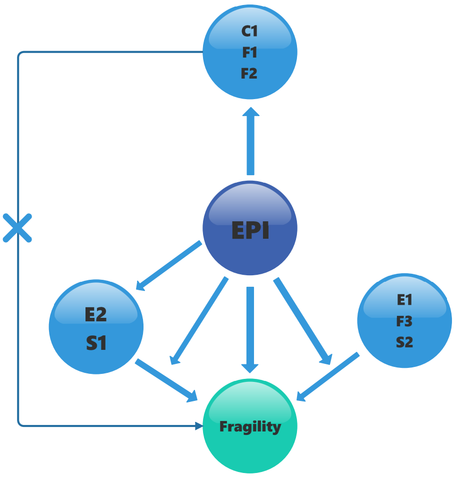


Figure 6: Illustration of Effects of Environmental Performance when $\mathrm{EPI} > 50$

There is no other variable that appear individually; therefore, we are confident that EPI act as a moderator between human factors and fragility.

To conclude, EPI moderates the relation of these following human factors:

- Economic Decline
- Uneven Economic Development
- Human Rights and Rule of Law
- Refugees and IDPs

Indeed, environmental performance does not have a clear and direct relationship with these variables. The moderator effect of environment on many economic indicators can be explained by Kuznets curve [14, 15].

Mediator Effect EPI's direct impact on fragility is already verified in Table 5 and Table 2, since predicting fragility using EPI alone would result in a even higher level of significance as in the model of Table 2.

We verify the role of human factors by establishing regression models in which EPI is used to predict each individual indicator of FSI. When EPI significantly predicts an indicator, we establish another regression model, in which EPI and the indicator jointly predict fragility.

These following indicators are mediator variables, through which environmental factors indirectly exhibit influence on fragility:

- Public Services
- Demographic Pressures

Clearly, environmental factors directly influences a state's public services and demographic pressures. The influence then propagates to fragility.

When the Index $\mathrm{EPI} < 50$

These states are environmentally fragile, and fragility of these states is no longer sensitive to environmental fragility.

Mediator Effect. We establish a regression model, using EPI to predict fragility, for states with EPI< 50. However, the coefficient of EPI in this case is -0.0003162, with p-value being 0.225. EPI is insignificant, which is also verified in Figure 4(a). Since environment does not influence fragility directly, mediator effect is impossible.

Var.	Coef.	P-val.	Level
C1	6.174e-4	0.033153	Medium
C2	-1.334e-3	0.001708	High
C3	6.624e-3	0.001896	High
E1	9.62e-3	0.004577	High
E3	-4.759e-3	1.63e-14	Super
P1	8.484e-3	0.006177	High
P3	-1.391e-2	0.000802	High
S2	2.320e-3	5.95e-8	Super
X1	7.863e-4	0.005399	High
EPI×C3	-1.68e-4	0.000992	Medium
EPI×P3	3.123e-4	0.000822	Medium

Table 3: Moderator Effect when EPI< 50

Moderator Effect. The result of moderator effect when EPI< 50 is in Table 3.

Although the state is fragile, and its fragility is insensitive to environmental change, environments moderates the impact of two human factors, C3 and P3, as clearly demonstrated in the table. C3 and P3 are respectively Group Grievance and Human Rights and Rule of Law.

3.4 Case Study

A Fragile State: Iraq. Iraq is, by FSI, one of the 10 most fragile states in the world. Also, the EPI Index of Iraq is below 50, the tipping point for environment performance. Using regression model in Eqn. 14 and coefficients in Table 3, we calculate the fragility score of Iraq to be 0.962, while the fragility score of Iraq calculated in Section 3.2 is 0.9615. The low residual verifies the efficiency of our model for environmentally fragile states.

Utilizing PSM, we find that the direct impact of environmental fragility on FSI score of Iraq is -0.075.

By taking partial derivative, we obtain

$$\frac{\partial s_f}{\partial \mathrm{EPI}} = -1.68 \times 10^{-4}\mathrm{C3} + 3.123 \times 10^{-4}\mathrm{P3}$$

Where C3 is for Group Grievance, and P3 is for Human Rights and Rule of Law. Plugging in the values of corresponding indicators, we obtain that

$$\frac{\partial s_f}{\partial \mathrm{EPI}} = -1.6128 \times 10^{-3} + 2.7170 \times 10^{-3} = 0.0075$$

Combining direct and indirect effect, we see that decreasing EPI leads to increase in fragility of Iraq. Bad environment leads to more fragile state.

We may further investigate the moderator effect of EPI. If we see EPI as a constant, the coefficient of C3 can be seen as $-1.68e-4\text{EPI}+6.624\text{e}-3$, and the coefficient of P3 is $3.123e-4\text{EPI}-1.391\text{e}-2$. That is to say, decreasing EPI would amplify the effect of Group Grievance and Human Rights and Rule of Law.

To explain, Group Grievance focuses on divisions and schisms between different groups in society, and is inclуced by inequal distribution of resources, divisions, and communal violence. Since further deterioration of environmental performance in a fragile state would deplete resources, such effect is expectable. This also creates human rights issues, since bad environment in this case could even harm people's basic right of living.

A Stable State: Mauritius Mauritius is a country whose EPI is larger than the threshold 50, and is not in the list of top ten vulnerable in terms of FSI index. The fragility score is 0.315, confirming that it is comparably stable.

The direct effect calculated by PSM is -0.047. As we analyzed, environmental performance's indirect effect is exhibited both as a moderator variable and a mediator variable.

By similar analysis in the last part, we find that better environmental performance alleviates the pressures of human rights (P3), uneven economic development (E2), and economic decline(E1).

Unlike for environmentally fragile states, the moderator effect takes place. For Mauritius, increase of environmental performance would improve its public services and alleviate its demographic pressures.

3.5 Forecasting Environmental Performance

The temporal prescribed in Section 2.5 is implemented by these following steps: **Variables Preparation.** GDP growth rate and GDP (constant 2010 US) are chosen as

basis for our temporal model, and can be found on the data bank of the World Bank [13]. We denote GDP at time t by G_t and its growth rate $R_t \triangleq \frac{\Delta G_t}{\Delta t} = \frac{G_t - G_{t-1}}{G_{t-1}}$. E_t is the EPI index in time t. Oftentimes variables are scaled, i.e first centralized by its mean, and normalized by its standard variance. The scaled version of these variables are written as $\overline{G_t}$, $\overline{R_t}$, and $\overline{E_t}$, while the scaling is performed according to the set of data at all time steps prior to and including t.

Forcasting GDP. The GDP Growth is modeled by ARIMA in Section 2.5. Furthermore, we choose three values of μ to represent different speed of economic development: $\mu = 0$ for stable, $\mu = 0.3$ for moderate, and $\mu = 0.5$ for fast. The prediction of GDP growth rate in these three settings is in Figure 10(a).

Predicting EPI. Now every variable is well prepared for the prediction of EPI index, as in Figure 10(b). The specifics are as below. The temporal equation is expressed as:

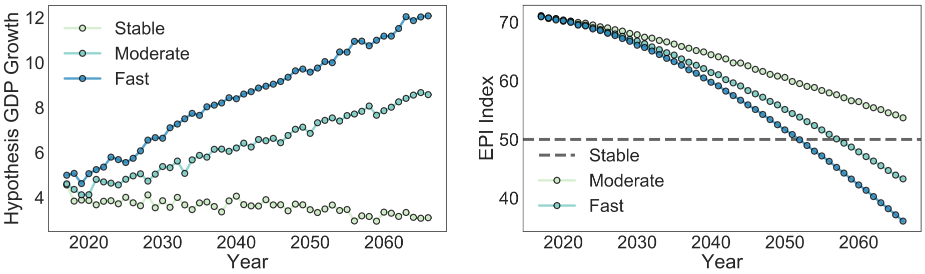
$$E_t = \beta_0 E_{t-1} + \beta_1 \log(G_t) + \beta_2 R_t + \beta_3 \overline{R_t} \times \overline{G_t} + \beta_4 \overline{R_t} \times \overline{R_t} \tag{16}$$

The parameters are determined by standard linear regression, with resutls listed in Figure 4. The signs of coefficients show the effects of variables:

Variable	E_{t-1}	$\log(G_t)$	R_t	$\overline{E_{t-1}} \times \overline{G_t}$	$\overline{E_{t-1}} \times \overline{R_t}$
Coefficient	0.35	-6.79	-0.21	-0.71	0.07

Table 4: Parameters for our Temporal Model.

- Good environmental performance tends to become better;
- Fast GDP Growth tends to damage environmental performance; however, when EPI is sufficiently high, the damage is neutralized.



(a) Future GDP Growth Rate

(b) Future EPI Index

Figure 7: Future GDP and EPI Index. Each is predicted under three different development settings: fast economic growth, moderate economic growth, and stable economic growth.

3.6 Trade-off between Development and Environment

From Figure 10(b), we see that in stable development setting, Mauritius does not reach a environmental fragile status within the range of 50 years.

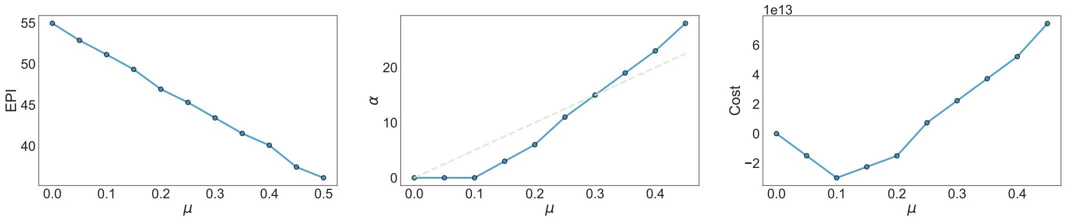
However, faster economic development brings instablizing factors. Under moderate and fast development settings, where μ is 0.3 and 0.5 respectively, Mauritius would be environmentally fragile within 50 years. The faster the development, the sooner it becomes fragile.

We then consider another hyperparameter α proposed in Section 2.5, used to represent the investment of a country to control for environmental instability possibly brought by economic development.

Remember that α does not change the predicted value of G_t, R_t per se. Instead, when predicting EPI, the values of variables used in Eqn. 16 is adjusted: $R_t \leftarrow R_t - \alpha$, $G_t \leftarrow$

$(R_t - \alpha)G_{t-1} + G_{t-1}$. Therefore, it is considered the annual governmental investment as percentage of GDP to improve environmental performance.

We visualized the minimum α, i.e. percentage of GDP as investment to protect environment, that is required to guaranteed that the state's EPI is larger than 50 in 50 years. The result is as in Figure 8.



(a) EPI after 50 years (b) Minimum Investment (c) Total Cost

Figure 8: Trade-off between Development and Environment. In 8(a) the data are EPI after 50 years. In 8(b), the dotted line is the lowest α to make sure fragility is stable after 50 years. The dashed line is 50μ, the theoretical contribution of fast economic growth after 50 years. 8(c) plots the total cost of environmental intervention in 50 years in terms of US dollar in 2010 with minimum α in 8(b).

Clearly, when the economic development is sufficiently large, the government need to invest a certain amount of money to ensure environmental stability.

We are interested in the cost of environmental protection. The investment is calculated as αG_t for each year t. For simplicity, we assume that the state gains $t\mu G_t$ for each year t for fast economic development, since theoretically, the economic development contributes accumulatively μ percent of GDP growth each year.

Therefore, annual cost of environmental protection is $(\alpha - t\mu)G_t$.

Figure 8 shows a trade-off between fast economic development and environmental protection.

A moderate rate of development, such as 0.3, would make sure that the benefit of economic growth is at least able to compensate most of the loss in environmental performance. For example, when $\mu = 0.3, \alpha = 15$, the total cost in 50 years would be 22269423781162.1 US dollars (constant 2010).

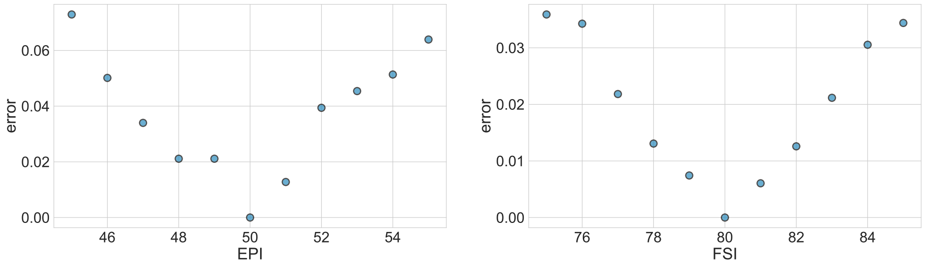
Smaller μ would even result in positive revenue in the range of 50 years. Indeed, the cost would exceed the contributino of μ in the first few years, but $t\mu$ exceeds α with certain $t < 50$. Therefore, such economic development pattern is ideal, since it achieves long-term balance of development and environmental protection.

4 Discussion

4.1 Parameter Sensitivity

1. **Sensitivity analysis of fragility against thresholds**

   First, we try to fine-tune the demarcation point of the EPI and FSI index and calculate the average of the absolute values of residuals. It turns out that there is not a significant deviation in terms of fragility.

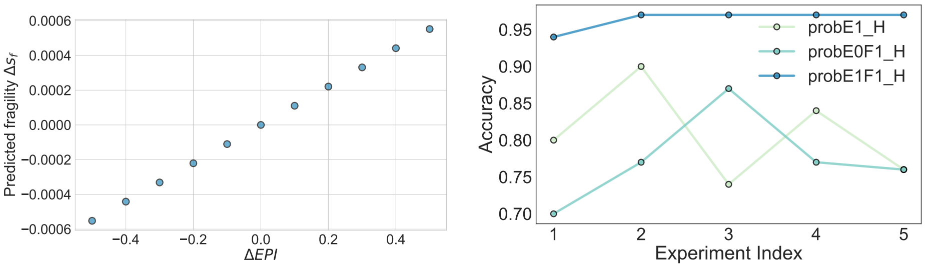


(a) Sensitivity against EPI (b) Sensitivity against FSI

Figure 9: Fragility Sensitivity Analysis

2. **Predicted model sensitivity analysis**

   Here we choose one country (Mauritius), the impact of the direct and indirect effects on the national fragile index are integrated. It is observed that there will not be a significant change in the predicted fragility when the EPI of the environmental index changes.



(a) Predicted Fragility Sensitivity Analysis (b) Cross Validation

Figure 10: Sensitivity Analysis

3. **Cross-validation for logistic regression**

   When calculating the fragility, we use logistic regression to estimate probability. Here we test the accuracy of logistic regression with 5-fold cross-validation. The regression accuracy of different probability varies as the figure shows:

4.2 Strength and Weakness

We first give overall strength and weakness for our entire framework:

- **Simplicity**. Our model is dedicated to capture the interaction between human and environmental factors in a simple, explainable, and verifiable fasion.
- **Objective**. Basic ideas are derived from convincing facts, while tools used to implement our ideas are theoretically well supported. Human error can be reduced to minimum.
- **Flexibility**. Our theoretical framework applies to many scenarios other than countries and can be implemented in various methods.
- **Appealing Results**. The result of our statistical analysis yields concrete conclusions consistent with interdisciplinary facts. The simulation results are convincing and insightful in finding the trade-off between fast economic growth and environment.

However, there are plenty of rooms for improvement for our model.

- Some hypothesis are too ideal. For example, we assumed that GDP is representative of human factors in the temporal model, and that EPI represents environmental fragility, without modeling the complex dynamics of climate change.
- Data limitation. Our simulation suffers from insufficient amount of data.
- Insufficient long-term prediction. Our long-term prediction of GDP and EPI used only recent information of a particular state and cannot well recognize complex dynamics in the long run.

4.3 Extending the Model

For smaller "states" such as cities, our theoretical framework still applies. However, specific data preparation, variable selection, and model implementation should be modified accordingly. For larger "states" such as continents, our model is in need of describing the interactive relationship between countries to capture the evolution of the climate change and fragility of a organized whole, and its effect on each individual component state.

We give some analysis as follows:

- External Intervention: It is one of elements given in the FSI index, which in turn plays an important role in our model. Here if we apply our model to figure out issues in terms of regions of various "scale", there is no doubt that its influences will be significantly different. For example, cities of a country must have a closely connection with each other, while continents are not likely to be the same, or certainly not "that closely". One way to solve this problem is that we can set some systematic measures to represent it. Actually, we can even take it out to discuss its impact with FSI and EPI if necessary.

- Different Functions of Cities and National Conditions: This is especially required to be considered when our model is applied to smaller "states". Different Functions of cities means that fragility is not simply evaluated based on some indexes when it comes to smaller "states" issues. Instead, we are supposed to think over its function in this country. For example, some cities with better economy development or important political status may still not fragile although they have a terrible environment. Besides, different national condition will also affect the model, like different powers of government in every country. To propose a better model, we may need to cluster different regions and describe the connection in every region more reasonably.
- Internal Difference of Continents: Continents, to the contrary, are so few that there is little variety all over the world. However, if we would like to tell which continent is more fragile, it is necessary to distinguish its internal difference. For example, the fragility of different regions in Asia probably range a lot.

4.4 Relation to Environmental Science

Peter Schwartz et al. [3] created an abrupt climate change scenario and analyzed its influence on United States National Security. Although its imagine was reasonable and meaningful for US society, the model was a little simple considering that it was one of pioneer works focusing on environmental security. Instead, our model pay more attention on the relation among human factors, environmental factors and fragility of states.

Like the former work, Ole Magnus Theisen et al. [4] did research on the history of climate change and conflict occurrence. It contributed to linking climate change to conflict, and its methodology of linking elements is similar to our studies of correlation among independent factors. However, our model is interested in the whole framework of relation rather than some concrete climate changes.

Amy Richmond Krakowka et al. [5] revealed the environmental security model in an Africa scenario. Its model mainly came from the relation among environment, security and conflict. With a process-outcome framework, it derived a specific conflict analysis, but our model concentrate on the global situation. Besides, we also consider the temporal model. In the process, we figure out how to take measures for the trade-off between economy development and environment protection.

5 Conclusion

We proposed a novel probabilistic framework to evaluate a state's fragility. Based on this, we analyzed the direct and indirect of environmental factors on fragility, and the interaction between environmental and human factors in their joint contribution to fragility.

We also proposed a temporal framework to forecase future environmental change. Our model found that moderate economic growth balances the trade-off between fast economic development and environmental damage in the long term.

References

[1] "Fragile states index," http://fundforpeace.org/fsi/, accessed:2018-02-09.

[2] "Fragile states index," https://epi.envirocenter.yale.edu/, accessed:2018-02-09.

[3] P. Schwartz and D. Randall, "An abrupt climate change scenario and its implications for united states national security," CALIFORNIA INST OF TECH PASADENA JET PROPULSION LAB, Tech. Rep., 2003.

[4] O. M. Theisen, N. P. Gleditsch, and H. Buhaug, "Is climate change a driver of armed conflict?" *Climatic change*, vol. 117, no. 3, pp. 613–625, 2013.

[5] A. R. Krakowka, N. Heimel, and F. A. Galgano, "Modeling environmental security in sub-saharan africa," *The Geographical Bulletin*, vol. 53, no. 1, p. 21, 2012.

[6] M. Caliendo and S. Kopeinig, "Some practical guidance for the implementation of propensity score matching," *Journal of economic surveys*, vol. 22, no. 1, pp. 31–72, 2008.

[7] R. H. Dehejia and S. Wahba, "Propensity score-matching methods for nonexperimental causal studies," *Review of Economics and statistics*, vol. 84, no. 1, pp. 151–161, 2002.

[8] K. Hirano and G. W. Imbens, "The propensity score with continuous treatments," *Applied Bayesian modeling and causal inference from incomplete-data perspectives*, vol. 226164, pp. 73–84, 2004.

[9] A. C. Ian Goodfellow, Yoshua Bengio, "Deep learning," 2016.

[10] R. M. Baron and D. A. Kenny, "The moderator–mediator variable distinction in social psychological research: Conceptual, strategic, and statistical considerations." *Journal of personality and social psychology*, vol. 51, no. 6, p. 1173, 1986.

[11] L. S. Aiken, S. G. West, and R. R. Reno, *Multiple regression: Testing and interpreting interactions*. Sage, 1991.

[12] J. D. Hamilton, *Time series analysis*. Princeton university press Princeton, 1994, vol. 2.

[13] “World bank open data,” https://data.worldbank.org/, accessed:2018-02-09.

[14] S. Kuznets, “Economic growth and income inequality,” *The American economic review*, pp. 1–28, 1955.

[15] S. Kuznets and J. T. Murphy, *Modern economic growth: Rate, structure, and spread*. Yale University Press New Haven, 1966, vol. 2.

Appendices

Appendix A

Example Python code:

```
# coding: utf-8

# In[140]:

import csv
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
import numpy as np
import matplotlib.pyplot as plt


# In[147]:

def read_fsi():
    csvFile = open("modified_fsi-2017.csv", "r")
    reader = csv.reader(csvFile)
    attr_dic = {}
    score_dic = {}
    attributes = []
    for item in reader:
        # ignore the first line
        if reader.line_num == 1:
            attributes = item[4:]
            continue
        attr_dic[item[0]] = [float(i) for i in item[4:]]
        score_dic[item[0]] = float(item[3])
    csvFile.close()
    return attr_dic, score_dic, attributes


# In[148]:

def input_fsi():
    attr_dic, score_dic, attributes = read_fsi()
    for (idx, item) in enumerate(attr_dic):
        if idx == 0:
            attr = np.array(attr_dic[item])
        new_attr = np.array(attr_dic[item])
        attr = np.vstack((attr, new_attr))
    for (idx, item) in enumerate(score_dic):
        if idx == 0:
            score = np.array(score_dic[item])
        new_score = np.array(score_dic[item])
        score = np.vstack((score, new_score))
    return attr, score


# In[153]:

def read_epi():
    csvFile = open("modified_EPI.csv", "r")
    reader = csv.reader(csvFile)
    attr_dic = {}
    score_dic = {}
    attributes = []
    for item in reader:
        # ignore the first line
        if reader.line_num == 1:
            attributes = item[3:10]
            continue
        attr_dic[item[2]] = [float(i) for i in item[3:10]]
        score_dic[item[2]] = float(item[10])
    csvFile.close()
    return attr_dic, score_dic, attributes


# In[154]:

def input_epi():
    attr_dic, score_dic, attributes = read_epi()
    for (idx, item) in enumerate(attr_dic):
        if idx == 0:
            attr = np.array(attr_dic[item])
        new_attr = np.array(attr_dic[item])
        attr = np.vstack((attr, new_attr))
    for (idx, item) in enumerate(score_dic):
        if idx == 0:
            score = np.array(score_dic[item])
        new_score = np.array(score_dic[item])
        score = np.vstack((score, new_score))
    return attr, score


# In[1126]:

def share_country():
    '''
    Find shared countries of FSI and EPI data.
    Return:
        FSI_data -- FSI data sorted according to countries.
        EPI_data -- EPI data sorted according to countries.
    Note that these two dictionaries have the same number of countries.
    And they also have the aligned attributes based on countries order.
    '''
    fsi_attrdic, fsi_scoredic, fsi_attributes = read_fsi()
    epi_attrdic, epi_scoredic, epi_attributes = read_epi()
    share_key = []
    cnt = 0
```

```
# First, delete all keys and values in fsi but not in epi
for (idx, item) in enumerate(fsi_attrdic):
    epi_key = list(epi_attrdic.keys())
    if item in epi_key:
        if cnt == 0:
            fsi_attr = np.array(fsi_attrdic[item]).reshape(1,-1)
            fsi_score = np.array(fsi_scoredic[item])
            share_key.append(item)
        else:
            share_key.append(item)
            fsi_attr = np.vstack((fsi_attr, fsi_attrdic[item]))
            fsi_score = np.vstack((fsi_score, fsi_scoredic[item]))
        cnt += 1
cnt = 0
# Second, delete all keys and values in epi but not in fsi
for (idx, item) in enumerate(epi_attrdic):
    fsi_key = list(fsi_attrdic.keys())
    if item in fsi_key:
        if cnt == 0:
            epi_attr = np.array(epi_attrdic[item]).reshape(1,-1)
            epi_score = np.array(epi_scoredic[item])
        else:
            epi_attr = np.vstack((epi_attr, epi_attrdic[item]))
            epi_score = np.vstack((epi_score, epi_scoredic[item]))
        cnt += 1
df = pd.DataFrame(data = {'country': share_key,
                          'FSI': fsi_score.ravel(),
                          'EPI': epi_score.ravel(),
                          fsi_attributes[0]: fsi_attr[:,0].ravel(),
                          fsi_attributes[1]: fsi_attr[:,1].ravel(),
                          fsi_attributes[2]: fsi_attr[:,2].ravel(),
                          fsi_attributes[3]: fsi_attr[:,3].ravel(),
                          fsi_attributes[4]: fsi_attr[:,4].ravel(),
                          fsi_attributes[5]: fsi_attr[:,5].ravel(),
                          fsi_attributes[6]: fsi_attr[:,6].ravel(),
                          fsi_attributes[7]: fsi_attr[:,7].ravel(),
                          fsi_attributes[8]: fsi_attr[:,8].ravel(),
                          fsi_attributes[9]: fsi_attr[:,9].ravel(),
                          fsi_attributes[10]: fsi_attr[:,10].ravel(),
                          fsi_attributes[11]: fsi_attr[:,11].ravel(),
                          epi_attributes[0]: epi_attr[:,0].ravel(),
                          epi_attributes[1]: epi_attr[:,1].ravel(),
                          epi_attributes[2]: epi_attr[:,2].ravel(),
                          epi_attributes[3]: epi_attr[:,3].ravel(),
                          epi_attributes[4]: epi_attr[:,4].ravel(),
                          epi_attributes[5]: epi_attr[:,5].ravel(),
                          epi_attributes[6]: epi_attr[:,6].ravel()})
df.to_csv('shared_data.csv')
fsi_df = pd.DataFrame(data = {'country': share_key,
                              'FSI': fsi_score.ravel(),
                              fsi_attributes[0]: fsi_attr[:,0].ravel(),
                              fsi_attributes[1]: fsi_attr[:,1].ravel(),
                              fsi_attributes[2]: fsi_attr[:,2].ravel(),
                              fsi_attributes[3]: fsi_attr[:,3].ravel(),
                              fsi_attributes[4]: fsi_attr[:,4].ravel(),
                                  fsi_attributes[5]: fsi_attr[:,5].ravel(),
                                  fsi_attributes[6]: fsi_attr[:,6].ravel(),
                                  fsi_attributes[7]: fsi_attr[:,7].ravel(),
                                  fsi_attributes[8]: fsi_attr[:,8].ravel(),
                                  fsi_attributes[9]: fsi_attr[:,9].ravel(),
                                  fsi_attributes[10]: fsi_attr[:,10].ravel(),
                                  fsi_attributes[11]: fsi_attr[:,11].ravel()
                                  })
    fsi_df.to_csv(’shared_fsi.csv’)
    epi_df = pd.DataFrame(data = {’country’: share_key,
                                  ’EPI’: epi_score.ravel(),
                                  epi_attributes[0]: epi_attr[:,0].ravel(),
                                  epi_attributes[1]: epi_attr[:,1].ravel(),
                                  epi_attributes[2]: epi_attr[:,2].ravel(),
                                  epi_attributes[3]: epi_attr[:,3].ravel(),
                                  epi_attributes[4]: epi_attr[:,4].ravel(),
                                  epi_attributes[5]: epi_attr[:,5].ravel(),
                                  epi_attributes[6]: epi_attr[:,6].ravel()
                                  })
    epi_df.to_csv(’shared_epi.csv’)
    return share_key, fsi_attr, epi_attr


# In[1134]:

country, fsi_attr, epi_attr = share_country()


# In[1135]:

def input_result():
    csvFile = open("shared_data.csv", "r")
    reader = csv.reader(csvFile)
    fsi_dic = {}
    epi_dic = {}
    fsi = []
    epi = []
    for item in reader:
        # ignore the first line
        if reader.line_num == 1:
            continue
        fsi_dic[item[1]] = float(item[2])
        epi_dic[item[1]] = float(item[3])
        fsi.append(float(item[2]))
        epi.append(float(item[3]))
    csvFile.close()
    return fsi, epi


# In[1137]:

fsi, epi = input_result()


# In[1138]:
```

```
plt.scatter(fsi, epi)
plt.show()


# In[604]:

from sklearn.cluster import KMeans


# In[605]:

def cluster(data, n_clusters):
    '''
    Cluster data into n_clusters categories.
    Input:
        data -- The input data to be clustered.
        n_clusters -- The number of categories to be clustered.
    Return:
        label_pred -- The category array corresponding to every data.
    '''
    estimator = KMeans(n_clusters = n_clusters)
    estimator.fit(data)
    label_pred = estimator.labels_
    print(label_pred)
    return label_pred


# In[1139]:

FSI_data = np.array([fsi]).reshape([-1,1])
EPI_data = np.array([epi]).reshape([-1,1])
#FSI_label = cluster(FSI_data, 2)
#EPI_label = cluster(EPI_data, 2)
FSI_label = (FSI_data>80).ravel()
EPI_label = (EPI_data>50).ravel()
plt.scatter(FSI_data[FSI_label == 1], EPI_data[FSI_label == 1], c = 'r')
plt.scatter(FSI_data[FSI_label == 0], EPI_data[FSI_label == 0], c = 'b')
plt.show()


# In[1140]:

plt.scatter(FSI_data[EPI_label == 1], EPI_data[EPI_label == 1], c = 'r')
plt.scatter(FSI_data[EPI_label == 0], EPI_data[EPI_label == 0], c = 'b')
plt.show()


# In[1141]:

def find_bound(data, label, mode = 1):
    '''
    Find the boundary of clusters.
    Input:
        data -- The input data to be clustered.
        label -- The label corresponding to the input data.
        mode -- mode = 1 means we need to find the minimum bound of data with label =
                mode = 0 means we need to find the maximum bound of data with label =
    Return:
        bound -- The boundary of 1-dim cluster.
    '''
    if mode:
        label1_min = min(data[label == 1])
        label0_max = max(data[label == 0])
        bound = (label1_min+label0_max)/2
    else:
        label1_max = max(data[label == 1])
        label0_min = min(data[label == 0])
        bound = (label1_max+label0_min)/2
    return bound


# In[1142]:

FSI_bound = find_bound(FSI_data, FSI_label)
EPI_bound = find_bound(EPI_data, EPI_label)
plt.scatter(FSI_data[(FSI_label == 1) & (EPI_label == 1)], EPI_data[(FSI_label == 1)
plt.scatter(FSI_data[(FSI_label == 1) & (EPI_label == 0)], EPI_data[(FSI_label == 1)
plt.scatter(FSI_data[(FSI_label == 0) & (EPI_label == 1)], EPI_data[(FSI_label == 0)
plt.scatter(FSI_data[(FSI_label == 0) & (EPI_label == 0)], EPI_data[(FSI_label == 0)
FSI_bound_y = np.linspace(min(EPI_data), max(EPI_data), 1000)
FSI_bound_x = np.full(FSI_bound_y.shape, FSI_bound)
EPI_bound_x = np.linspace(min(FSI_data), max(FSI_data), 1000)
EPI_bound_y = np.full(EPI_bound_x.shape, EPI_bound)
plt.plot(FSI_bound_x, FSI_bound_y, c = 'k')
plt.plot(EPI_bound_x, EPI_bound_y, c = 'k')
plt.xlabel('FSI')
plt.ylabel('EPI')
plt.show()


# In[725]:

from mlxtend.plotting import plot_decision_regions


# In[774]:

def logistic_util(X, Y, test_size, random_state):
    '''
    Implement logistic regression with sklearn.
    Input:
        X -- The independent variables set.
        Y -- The dependent variable/target set.
        test_size -- Test size of the whole set.
        random_state -- Param for train_test_split.
    Output:
        lr -- Logistic Regression Result.
    '''
    #X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = test_size,
    #Y_train, Y_test = Y_train.ravel(), Y_test.ravel()
    #X_train, Y_train = train_test_split(X, Y, test_size = test_size, random_state =
    #Y_train = Y_train.ravel(), Y_test.ravel()
    #print(X_train.shape, X_test.shape, Y_train.shape, Y_test.shape)
    X_train = X
    Y_train = Y
    # Normalize the data
    sc = StandardScaler()
    sc.fit(X_train)
    X_train_std = np.array(list(sc.transform(X_train)))
    #X_test_std = np.array(list(sc.transform(X_test)))
    # Combine the train and test data
    #X_combined_std = np.vstack((X_train_std, X_test_std))
    #Y_combined = np.hstack((Y_train, Y_test))
    # Implement logistic regression
    lr = LogisticRegression(C = 0.01, random_state = 0)
    lr.fit(X_train_std, Y_train)
    #print(lr.predict_proba(X_test_std[2,:].reshape(1,-1))[0,0], Y_test[2])
    scores = cross_val_score(lr, X, Y, cv = 5, scoring = 'accuracy')
    print(scores)
    return lr, X_train_std, Y_train
    '''
    plot_decision_regions(X_combined_std, Y_combined, clf = lr, res = 0.02)
    plt.xlabel('petal length')
    plt.ylabel('petal width')
    plt.legend(loc = 'upper left')
    plt.show()
    '''


# In[775]:

from sklearn.cross_validation import cross_val_score


# In[1167]:

print(fsi_attr[0])


# In[1143]:

lr1, fsi_attr_std, _ = logistic_util(fsi_attr, EPI_label, 0, 0)
probE0_H = lr1.predict_proba(fsi_attr_std)[:,0]
probE1_H = np.ones(probE0_H.shape)-probE0_H


# In[1144]:

lr2, fsi_attr_std, _ = logistic_util(fsi_attr, (EPI_label == 0) & (FSI_label == 1), 0
prob_other = lr2.predict_proba(fsi_attr_std)[:,0]
probE0F1_H = np.ones(prob_other.shape)-prob_other


# In[1145]:
```

```
lr3, fsi_attr_std, _ = logistic_util(fsi_attr, (EPI_label == 1) & (FSI_label == 1), 0
prob_other = lr3.predict_proba(fsi_attr_std)[:,0]
probE1F1_H = np.ones(prob_other.shape)-prob_other


# In[1146]:

probF1_E0H = probE0F1_H / probE0_H
probF1_E1H = probE1F1_H / probE1_H
print(probE0F1_H[EPI_label == 0], probE0_H[EPI_label == 0])


# In[1147]:

fragile_value = []
for i in range(len(fsi_attr_std)):
    if EPI_label[i] == 1:
        fragile_value.append(probF1_E1H[i])
    else:
        fragile_value.append(probF1_E0H[i])


# In[1148]:

fragile = np.array(fragile_value)
country = np.array(country)
data_num = len(fragile_value)
df_fragile = pd.DataFrame(data = {’country’: country.reshape(data_num),
                                  ’fragile’: fragile.reshape(data_num),
                                  ’FSI’: FSI_data.reshape(data_num),
                                  ’EPI’: EPI_data.reshape(data_num)})


# In[1149]:

df_fragile.sort_values(’fragile’,axis = 0,ascending = ’False’)
df_fragile.to_csv(’fragile.csv’)


# In[736]:

fragile_value = np.array(fragile_value).reshape(-1,1)
sc = StandardScaler()
sc.fit(fragile_value)
fragile_std = (255*(np.array(list(sc.transform(fragile_value)))+1)/2).astype(np.uint8
fragile_std = fragile_std.reshape(len(fragile_std))
print(fragile_std)


# In[578]:

fragile_value = np.array(fragile_value)
#fragile_value = np.tanh(fragile_value)
stretch = int(255/max(fragile_value))
```

```
fragile_std = (fragile_value*stretch).astype(np.uint8)
fragile_std = fragile_std.reshape(len(fragile_std))
print(fragile_std)


# In[1150]:

fragile = np.array(fragile_value)
sort_index = fragile.argsort()


# In[1151]:

import seaborn as sn
pal = sn.color_palette("GnBu_d", 149)


# In[1152]:

color = []
for i in range(len(fragile_value)):
    color.append(pal[sort_index[i]])


# In[1153]:

FSI_bound = find_bound(FSI_data, FSI_label)
EPI_bound = find_bound(EPI_data, EPI_label)
plt.scatter(FSI_data[(FSI_label == 1) & (EPI_label == 1)], EPI_data[(FSI_label == 1)
plt.scatter(FSI_data[(FSI_label == 1) & (EPI_label == 0)], EPI_data[(FSI_label == 1)
plt.scatter(FSI_data[(FSI_label == 0) & (EPI_label == 1)], EPI_data[(FSI_label == 0)
plt.scatter(FSI_data[(FSI_label == 0) & (EPI_label == 0)], EPI_data[(FSI_label == 0)
FSI_bound_y = np.linspace(min(EPI_data), max(EPI_data), 1000)
FSI_bound_x = np.full(FSI_bound_y.shape, FSI_bound)
EPI_bound_x = np.linspace(min(FSI_data), max(FSI_data), 1000)
EPI_bound_y = np.full(EPI_bound_x.shape, EPI_bound)
plt.plot(FSI_bound_x, FSI_bound_y, c = ’k’)
plt.plot(EPI_bound_x, EPI_bound_y, c = ’k’)
plt.xlabel(’FSI’)
plt.ylabel(’EPI’)
plt.show()


# In[404]:

print(probE1_H[129])
print(probE1F1_H[129])
print(probF1_E1H[129], fragile_value[129])


# In[422]:

print(probE1_H[10], EPI_label[10])
```

```
# In[1154]:

def reverse_num(FSI_data, EPI_data, fragile):
    cnt = 0
    rate = []
    for i in range(len(FSI_data)):
        mask = (EPI_data > EPI_data[i]+5) & (FSI_data+1 < FSI_data[i]) & (fragile > f
        rate.append(1.0*sum(mask)/(len(FSI_data)-1))
    avg_rate = np.mean(rate)
    print(avg_rate)


# In[1155]:

n_samples = len(FSI_data)
reverse_num(FSI_data.reshape(n_samples), EPI_data.reshape(n_samples), fragile.reshape


# In[844]:

from sklearn import linear_model
import math
import sys


# In[1119]:

#  implementation of propensity score matching algorithm
#  input:
#   feats: m * n array
#   label: m * 1 binary array, 1 -> treated label, 0 -> untreated label
#   lb : m * 1 binary array, labels of samples, 1 -> positive sample, 0 -> negative s
#   trim : leave the samples which have the probability in [trim, 1-trim]
#  output:
#   pos : number of positive samples after matching
#   neg : number of negative samples after matching
#   pre_indic : m * 1 array, std bias of treatment and untreatment before matching,
#       each dimension is correpondent to a feature.
#   post_indic : m * 1 array, std bias of each feature after matching

def propensity_score_matching(feats,label, lb, trim=0.01):
        concernedidx = np.array(range(len(lb)))[label == 1]
        to_matched = np.array(range(len(lb)))[label != 1]

        clf = linear_model.LogisticRegression(solver='sag', max_iter=5000)
        #print(clf.fit(feats, label))
        clf.fit(feats,label)
        predict_proba = clf.predict_proba(feats)[:,1]

        overlap_range_min = trim
        overlap_range_max = 1-trim
        # print overlap_range_min,overlap_range_max,'\r'
        concerned_value = {s:predict_proba[s] for s in concernedidx if overlap_range_
        match_value = {s:predict_proba[s] for s in to_matched if overlap_range_min<=p
        concerned_value = list(sorted(concerned_value.items(), key=lambda x:x[1]))
        match_value = list(sorted(match_value.items(), key=lambda x:x[1]))

        pairs = []
        curpos = 0
        for k, value in enumerate(concerned_value):
            if k % 1000 == 0:
                sys.stdout.write(’\r{}/{} nodes processed’.format(k,len(concerned_val
                sys.stdout.flush()
            idx, prob = value
            while curpos < len(match_value) and prob > match_value[curpos][1]:
                curpos += 1
            if curpos == 0:
                tmp = 0
            elif curpos == len(match_value):
                tmp = curpos - 1
            else:
                tmp = curpos -1 if math.fabs(match_value[curpos-1][1] - prob) < math.
            pairs.append([idx, match_value[tmp][0]])
        pairs = np.array(pairs)
        minus = np.array([predict_proba[a] - predict_proba[b] for a,b in pairs])
        sigma = np.std(minus)
        selectedpair = pairs[np.abs(minus) < 2*sigma]
        treated = selectedpair[:,0]
        untreated = selectedpair[:,1]
        #print(’\rleft {}/{}’.format(len(selectedpair),len(concerned_value)))

        pre_tr_ft = feats[concernedidx]
        pre_ut_ft = feats[to_matched]
        pre_indic = (np.mean(pre_tr_ft, axis=0)-np.mean(pre_ut_ft, axis=0))
/(np.sqrt(0.5*(np.var(pre_tr_ft,axis=0)+np.var(pre_ut_ft,axis=0))))
        pos_tr_ft = feats[treated]
        pos_ut_ft = feats[untreated]
        pos_indic = (np.mean(pos_tr_ft, axis=0)-np.mean(pos_ut_ft, axis=0))
/np.sqrt(0.5*(np.var(pos_tr_ft,axis=0)+np.var(pos_ut_ft,axis=0)))

        pos = np.sum(lb[treated])
        neg = np.sum(lb[untreated])
        print(’’)
        print("result:pos {}, neg {}, ratio: {}, std bias {}->{}".format(pos, neg, po
        return [treated, untreated, pre_indic, pos_indic, predict_proba]


# In[1156]:

treated, untreated, pre_indic, pos_indic, pro_score = propensity_score_matching(fsi_a
print(np.mean(fragile[untreated]-fragile[treated]))
diff = fragile[untreated]-fragile[treated]


# In[1157]:

plt.hist(diff)
np.std(diff)
plt.show()
```

```
# In[1158]:

plt.hist(pro_score[treated],alpha=0.7,bins=8)
plt.hist(pro_score[untreated],alpha=0.7,bins=8)
plt.show()


# In[959]:

F_E1 = np.mean(fragile[EPI_label == 1])
F_E0 = np.mean(fragile[EPI_label == 0])
F_diff = F_E0-F_E1
print(F_diff)


# In[960]:

def find_comp(treated, untreated, FSI_data, EPI_data, fragile):
    df = pd.DataFrame(data = {’country1’: list(df_fragile[’country’][treated]),
                              ’country2’: list(df_fragile[’country’][untreated]),
                              ’fragility1’: fragile[treated].flatten(),
                              ’fragility2’: fragile[untreated].flatten()})
    df.to_csv(’comparison.csv’)


# In[1159]:

find_comp(treated, untreated, FSI_data, EPI_data, fragile)


# In[1160]:

FSI_label = FSI_label.astype(np.uint8)
NEPI_label = (np.ones(EPI_label.shape)-EPI_label).astype(np.uint8)
label = NEPI_label | FSI_label
treated, untreated, pre_indic, pos_indic, pro_score = propensity_score_matching(fsi_a
print(np.mean(fragile[treated]-fragile[untreated]))
diff = fragile[treated]-fragile[untreated]


# In[1161]:

plt.hist(diff)
np.std(diff)
plt.show()


# In[1162]:

plt.hist(pro_score[treated],alpha=0.7,bins=5)
plt.hist(pro_score[untreated],alpha=0.7,bins=5)
plt.show()
```

```
# In[1010]:

def find_comp(treated, untreated, FSI_data, EPI_data, fragile):
    df = pd.DataFrame(data = {’country1’: list(df_fragile[’country’][treated]),
                              ’country2’: list(df_fragile[’country’][untreated]),
                              ’FSI1’: FSI_data[treated].flatten(),
                              ’FSI2’: FSI_data[untreated].flatten(),
                              ’EPI1’: EPI_data[treated].flatten(),
                              ’EPI2’: EPI_data[untreated].flatten(),
                              ’fragility1’: fragile[treated].flatten(),
                              ’fragility2’: fragile[untreated].flatten(),
                              ’pro_score’: pro_score[treated]-pro_score[untreated]})
    df.to_csv(’comparison.csv’)


# In[1163]:

find_comp(treated, untreated, FSI_data, EPI_data, fragile)


# In[1014]:

def read_allfsi():
    csvFile = open("all_fsi.csv", "r")
    reader = csv.reader(csvFile)
    attr_dic = {}
    score_dic = {}
    attributes = []
    for item in reader:
        # ignore the first line
        if reader.line_num == 1:
            attributes = item[4:]
            continue
        attr_dic[item[0]+item[1]] = [float(i) for i in item[4:]]
        score_dic[item[0]+item[1]] = float(item[3])
    csvFile.close()
    return attr_dic, score_dic, attributes


# In[1021]:

def read_allepi():
    csvFile = open("all_epi.csv", "r")
    reader = csv.reader(csvFile)
    score_dic = {}
    attributes = []
    for item in reader:
        # ignore the first line
        if reader.line_num == 1:
            attributes = [int(i) for i in item[2:]]
            continue
        for i in range(len(attributes)):
            score_dic[item[1]+str(attributes[i])] = float(item[2+i])
    csvFile.close()
    return score_dic, attributes
```

```
# In[1026]:

fsi_attrdic, fsi_scoredic, fsi_attributes = read_allfsi()


# In[1027]:

epi_scoredic, epi_attributes = read_allepi()


# In[1038]:

def share_country_backdata():
    '''
    Find shared countries of all the FSI and EPI data in the past years.
    Return:
        FSI_data -- FSI data sorted according to countries.
        EPI_data -- EPI data sorted according to countries.
    Note that these two dictionaries have the same number of countries.
    And they also have the aligned attributes based on countries order.
    '''
    fsi_attrdic, fsi_scoredic, fsi_attributes = read_allfsi()
    epi_scoredic, epi_attributes = read_allepi()
    share_key = []
    cnt = 0
    # First, delete all keys and values in fsi but not in epi
    for (idx, item) in enumerate(fsi_attrdic):
        epi_key = list(epi_scoredic.keys())
        if item in epi_key:
            if cnt == 0:
                fsi_attr = np.array(fsi_attrdic[item]).reshape(1,-1)
                fsi_score = np.array(fsi_scoredic[item])
                epi_score = np.array(epi_scoredic[item])
                share_key.append(item)
            else:
                share_key.append(item)
                fsi_attr = np.vstack((fsi_attr, fsi_attrdic[item]))
                fsi_score = np.vstack((fsi_score, fsi_scoredic[item]))
                epi_score = np.vstack((epi_score, epi_scoredic[item]))
            cnt += 1
    df = pd.DataFrame(data = {'country': share_key,
                              'FSI': fsi_score.ravel(),
                              'EPI': epi_score.ravel(),
                              fsi_attributes[0]: fsi_attr[:,0].ravel(),
                              fsi_attributes[1]: fsi_attr[:,1].ravel(),
                              fsi_attributes[2]: fsi_attr[:,2].ravel(),
                              fsi_attributes[3]: fsi_attr[:,3].ravel(),
                              fsi_attributes[4]: fsi_attr[:,4].ravel(),
                              fsi_attributes[5]: fsi_attr[:,5].ravel(),
                              fsi_attributes[6]: fsi_attr[:,6].ravel(),
                              fsi_attributes[7]: fsi_attr[:,7].ravel(),
                              fsi_attributes[8]: fsi_attr[:,8].ravel(),
                              fsi_attributes[9]: fsi_attr[:,9].ravel(),
                              fsi_attributes[10]: fsi_attr[:,10].ravel(),
                              fsi_attributes[11]: fsi_attr[:,11].ravel()\
                             })
    df.to_csv('shared_alldata.csv')
    fsi_df = pd.DataFrame(data = {'country': share_key,
                                  'FSI': fsi_score.ravel(),
                                  fsi_attributes[0]: fsi_attr[:,0].ravel(),
                                  fsi_attributes[1]: fsi_attr[:,1].ravel(),
                                  fsi_attributes[2]: fsi_attr[:,2].ravel(),
                                  fsi_attributes[3]: fsi_attr[:,3].ravel(),
                                  fsi_attributes[4]: fsi_attr[:,4].ravel(),
                                  fsi_attributes[5]: fsi_attr[:,5].ravel(),
                                  fsi_attributes[6]: fsi_attr[:,6].ravel(),
                                  fsi_attributes[7]: fsi_attr[:,7].ravel(),
                                  fsi_attributes[8]: fsi_attr[:,8].ravel(),
                                  fsi_attributes[9]: fsi_attr[:,9].ravel(),
                                  fsi_attributes[10]: fsi_attr[:,10].ravel(),
                                  fsi_attributes[11]: fsi_attr[:,11].ravel()
                                  })
    fsi_df.to_csv('shared_allfsi.csv')
    epi_df = pd.DataFrame(data = {'country': share_key,
                                  'EPI': epi_score.ravel()
                                  })
    epi_df.to_csv('shared_allepi.csv')
    return share_key, fsi_attr


# In[1040]:

share_key, fsi_attr = share_country_backdata()


# In[1041]:

def input_allresult():
    csvFile = open("shared_alldata.csv", "r")
    reader = csv.reader(csvFile)
    fsi_dic = {}
    epi_dic = {}
    fsi = []
    epi = []
    for item in reader:
        # ignore the first line
        if reader.line_num == 1:
            continue
        fsi_dic[item[1]] = float(item[2])
        epi_dic[item[1]] = float(item[3])
        fsi.append(float(item[2]))
        epi.append(float(item[3]))
    csvFile.close()
    return fsi, epi


# In[1042]:
```

```
fsi, epi = input_allresult()


# In[1096]:

FSI_data = np.array([fsi]).reshape([-1,1])
EPI_data = np.array([epi]).reshape([-1,1])
#FSI_label = cluster(FSI_data, 2)
#EPI_label = cluster(EPI_data, 2)
FSI_label = (FSI_data>84).ravel()
EPI_label = (EPI_data>60).ravel()
plt.scatter(FSI_data[FSI_label == 1], EPI_data[FSI_label == 1], c = 'r')
plt.scatter(FSI_data[FSI_label == 0], EPI_data[FSI_label == 0], c = 'b')
plt.show()


# In[1097]:

plt.scatter(FSI_data[EPI_label == 1], EPI_data[EPI_label == 1], c = 'r')
plt.scatter(FSI_data[EPI_label == 0], EPI_data[EPI_label == 0], c = 'b')
plt.show()


# In[1098]:

FSI_bound = find_bound(FSI_data, FSI_label)
EPI_bound = find_bound(EPI_data, EPI_label)
plt.scatter(FSI_data[(FSI_label == 1) & (EPI_label == 1)], EPI_data[(FSI_label == 1)
plt.scatter(FSI_data[(FSI_label == 1) & (EPI_label == 0)], EPI_data[(FSI_label == 1)
plt.scatter(FSI_data[(FSI_label == 0) & (EPI_label == 1)], EPI_data[(FSI_label == 0)
plt.scatter(FSI_data[(FSI_label == 0) & (EPI_label == 0)], EPI_data[(FSI_label == 0)
FSI_bound_y = np.linspace(min(EPI_data), max(EPI_data), 1000)
FSI_bound_x = np.full(FSI_bound_y.shape, FSI_bound)
EPI_bound_x = np.linspace(min(FSI_data), max(FSI_data), 1000)
EPI_bound_y = np.full(EPI_bound_x.shape, EPI_bound)
plt.plot(FSI_bound_x, FSI_bound_y, c = 'k')
plt.plot(EPI_bound_x, EPI_bound_y, c = 'k')
plt.xlabel('FSI')
plt.ylabel('EPI')
plt.show()


# In[1099]:

print(len(FSI_label[FSI_label==1])/len(FSI_label), len(EPI_label[EPI_label == 1])/len


# In[1100]:

lr1, fsi_attr_std, _ = logistic_util(fsi_attr, EPI_label, 0, 0)
probE0_H = lr1.predict_proba(fsi_attr_std)[:,0]
probE1_H = np.ones(probE0_H.shape)-probE0_H
lr2, fsi_attr_std, _ = logistic_util(fsi_attr, (EPI_label == 0) & (FSI_label == 1), 0
prob_other = lr2.predict_proba(fsi_attr_std)[:,0]
probE0F1_H = np.ones(prob_other.shape)-prob_other
```

```
lr3, fsi_attr_std, _ = logistic_util(fsi_attr, (EPI_label == 1) & (FSI_label == 1), 0
prob_other = lr3.predict_proba(fsi_attr_std)[:,0]
probE1F1_H = np.ones(prob_other.shape)-prob_other
probF1_E0H = probE0F1_H / probE0_H
probF1_E1H = probE1F1_H / probE1_H
print(probE0F1_H[EPI_label == 0], probE0_H[EPI_label == 0])


# In[1101]:

fragile_value = []
for i in range(len(fsi_attr_std)):
    if EPI_label[i] == 1:
        fragile_value.append(probF1_E1H[i])
    else:
        fragile_value.append(probF1_E0H[i])
fragile = np.array(fragile_value)
share_key = np.array(share_key)
data_num = len(fragile_value)
df_fragile = pd.DataFrame(data = {'country': share_key.reshape(data_num),
                                  'fragile': fragile.reshape(data_num),
                                  'FSI': FSI_data.reshape(data_num),
                                  'EPI': EPI_data.reshape(data_num)})


# In[1102]:

df_fragile.sort_values('fragile',axis = 0,ascending = 'False')
df_fragile.to_csv('fragile_all.csv')


# In[1103]:

print(data_num)


# In[1104]:

import seaborn as sn
pal = sn.color_palette("GnBu_d", data_num)
fragile = np.array(fragile_value)
sort_index = fragile.argsort()
color = []
for i in range(len(fragile_value)):
    color.append(pal[sort_index[i]])


# In[1105]:

FSI_bound = find_bound(FSI_data, FSI_label)
EPI_bound = find_bound(EPI_data, EPI_label)
plt.scatter(FSI_data[(FSI_label == 1) & (EPI_label == 1)], EPI_data[(FSI_label == 1)
plt.scatter(FSI_data[(FSI_label == 1) & (EPI_label == 0)], EPI_data[(FSI_label == 1)
plt.scatter(FSI_data[(FSI_label == 0) & (EPI_label == 1)], EPI_data[(FSI_label == 0)
plt.scatter(FSI_data[(FSI_label == 0) & (EPI_label == 0)], EPI_data[(FSI_label == 0)
FSI_bound_y = np.linspace(min(EPI_data), max(EPI_data), 1000)
FSI_bound_x = np.full(FSI_bound_y.shape, FSI_bound)
EPI_bound_x = np.linspace(min(FSI_data), max(FSI_data), 1000)
EPI_bound_y = np.full(EPI_bound_x.shape, EPI_bound)
plt.plot(FSI_bound_x, FSI_bound_y, c = 'k')
plt.plot(EPI_bound_x, EPI_bound_y, c = 'k')
plt.xlabel('FSI')
plt.ylabel('EPI')
plt.show()


# In[1106]:

n_samples = len(FSI_data)
reverse_num(FSI_data.reshape(n_samples), EPI_data.reshape(n_samples), fragile.reshape


# In[1120]:

FSI_label = FSI_label.astype(np.uint8)
NEPI_label = (np.ones(EPI_label.shape)-EPI_label).astype(np.uint8)
label = NEPI_label | FSI_label
treated, untreated, pre_indic, pos_indic, pro_score = propensity_score_matching(fsi_a
print(np.mean(fragile[treated]-fragile[untreated]))
diff = fragile[treated]-fragile[untreated]


# In[1121]:

plt.hist(diff)
np.std(diff)
plt.show()


# In[1123]:

plt.hist(pro_score[treated],alpha=0.7,bins=10)
plt.hist(pro_score[untreated],alpha=0.7,bins=10)
plt.show()


# In[1124]:

def find_comp(treated, untreated, FSI_data, EPI_data, fragile):
    df = pd.DataFrame(data = {'country1': list(df_fragile['country'][treated]),
                              'country2': list(df_fragile['country'][untreated]),
                              'FSI1': FSI_data[treated].flatten(),
                              'FSI2': FSI_data[untreated].flatten(),
                              'EPI1': EPI_data[treated].flatten(),
                              'EPI2': EPI_data[untreated].flatten(),
                              'fragility1': fragile[treated].flatten(),
                              'fragility2': fragile[untreated].flatten(),
                              'pro_score': pro_score[treated]-pro_score[untreated]})
    df.to_csv('comparison_all.csv')
```

```
# In[1125]:

find_comp(treated, untreated, FSI_data, EPI_data, fragile)


# In[ ]:
```