RDM Analytics

In this document you will find detailed information about how we define the Rural Development Model (RDM) profile and the methodology and different metrics used to calculate the RDM index. The former defines a profile of desirable characteristics that we would like suitable candidate sites to have, with the latter measuring this in a quantitative way. Both of these together form the basic structure of the RDM map, which is a map of suitable sites measured via the RDM index.

LAST DATE MODIFIED: 03/09/2020

RDM Profile

The Rural Development Model (RDM) profile is needed to assess the suitability of communities to the application of the RDM framework. More specifically, one needs to define a criteria that firstly identifies rural communities, and then secondly distinguishes between their potential to obtain self-sustainable growth. As a next step, we can then start looking for data and corresponding metrics which measure each aspect of the RDM profile that we aim to capture. We refer to communities that we are looking to apply our framework to as 'potential sites'.

The first concept in our RDM profile we must define is what we mean by a rural area. This definition can vary according to the national statistics agency. However, there is a *general understanding* that rural areas are those outside urban conglomerations i.e. towns and cities. This translates to areas of low population density.

Secondly, self sustainable growth is an important concept which encompases "meeting the needs of the present without compromising the needs of future generations to meet their own needs". Essentially, this underlies the need for long-term growth for potential sites. In practice, we want to maximise support to the environment and society, which are resources, whilst minimising the human and economic impact.

Variables of interest

- High unused clean energy resources. Must measure potential of each energy resource and the amount this has been exploited already. Ideally a potential site would have high availability that has been under exploited. We focus on these energy resources:
    - Solar
    - Wind

- Environmental quality. Potential sites must be less affected by pollutants which affect the quality of the environment.
- Population. Population density of potential sites should be sufficiently low. A highly densely populated area may not be suitable for our RDM framework.
- Infrastructure. Potential sites should have sufficient infrastructure in place. Since rural communities are very rarely served by public transport, this would involve assessing road accessibility. Sites with low/medium road accessibility are preferred since high road access is correlated with an already high development.
- Economic activity. There should be limited economic activity at potential sites. This is because sites which already have significant economic activity are likely to be already relatively more developed. This should also be linked relatively to poverty: lower economic activity will usually mean higher poverty levels.
- Social access.

Approach to Data Analytics Work

The aim is to measure the suitability of potential sites through an index that we call the RDM index. This will enable users to gain an understanding of the suitability of sites for the application of the RDM framework with one simple to understand number. The RDM index is a type of composite indicator i.e. a single number aggregated from the different metrics related to the RDM profile. We form the RDM index as a composite indicator because there are multiple factors we have already included - a single number allows us to summarise the insights from all of these factors. Additionally, it offers a flexible framework if in the future more data is added since then we just need to follow the same steps to produce an updated version of the RDM index. Examples of composite indicators which are used frequently include the Human Development Index (HDI).

One approach to form the RDM index is to simply aggregate the metrics purely based on theoretical understanding. However, this approach ignores any equally valuable information coming from the data itself. Therefore, the formation of the RDM index will be *data-driven* i.e. leveraging on data we can obtain that relates to the goals set out in the RDM profile. In this way we can combine a solid theoretical basis through the RDM profile with the natural advantages given in the data. In turn, this will be overlaid onto a map, giving an immediate, graphical way of identifying potential sites using the RDM index.

Below we describe the general approach and steps we will take in order to do this. Later on we shall define more specifically the details of each step in our context. of how we do this with the idea that it should be flexible enough in case new data is added. The step by step approach is:

1. Data Selection
2. Data Loading/Imputation
3. Normalisation/Preparation
4. Weighting/Aggregation
5. Uncertainty and sensitivity analysis
6. Visualization/Further Results

- Data Selection. Here we select the appropriate data that will be aggregated into the RDM index. This includes adding new data to a preexisting set that we already have.
  - Why? We can explore the strength of weaknesses of adding certain data.
  - How? Check quality of data - is it reliably sourced/calculated/frequently updated/recent? What metrics can we obtain from it? In the case that a new metric is added to a preexisting set, does it add any potential insight relative to the preexisting set and the theoretical framework? Is it of the same geographical scale?

- Normalisation/Preparation. Make metric values comparable through some normalisation procedure.
  - Why? Different metrics may have different scales, which means we cannot aggregate them without introducing bias simply from the different scales of the metrics.

    - How? Can simply z score, divide by min-max difference etc. Other more complicated normalisation procedures also exist. The different distributions of the metrics should be taken into account when designing a tailor made normalisation procedure.

- Data Imputation. Here we estimate missing values in the data.
    - Why? We have to do it to complete the dataset for use with some statistical algorithms that cannot deal with missing data. It will also give us a reliable way of checking the robustness of our final results.
    - How? Variety of methods ranging from the simple: replacing with a constant/average summary statistic to more complex such as Single Value Decomposition (SVD) to K Nearest Neighbours (KNN). This will be important when combining data at different geographical scales.

- Weighting/Aggregation. Deciding how we weigh and aggregate different metrics to produce the RDM index.
    - Why? Distill the same information offered by a set of metrics into one, easy to understand version.
    - How? Variety of methods exist from more generalisable such as PCA/unsupervised learning to tailor made schemes.

- Visualisation/Further results. Important to overlay the results onto a map so we can see graphically how they rank.
    - Why? Most succinct and visually informative way of presenting the results in this context.
    - How? Overlay results onto a map.

- Uncertainty analysis/Performance. Verify the effect of uncertainty/sensitivity in data.
    - Why? We can identify sources of uncertainty in the data and how this affects the data, and thus the end result. Measuring performance gives us an idea of how well the composite indicators are at summarising the multivariate structure.
    - How? The effect of uncertainty can be measured in this process by replacing/adding white noise, and comparing the final result to this new set. Create a benchmark and measure the results obtained to see if there is a positive difference.

Methodology/Metric formulations

Introduction

We want to form a singular RDM index which can summarise all the information coming from metrics derived from the datasets used. The RDM index is therefore a composite indicator which varies according to location.

Our strategy is to keep the calculation of the index simple so that it remains flexible enough to cope with the different number of datasets it is aggregated from. We propose to divide metrics according to categories specifically social, economic, geographic and environmental. This divides the metrics so that each category of metrics has a common aspect that will affect the RDM profile. We then need to aggregate metrics for each category, and finally take these new sets and aggregate these further to form the final RDM index. We outline the justification and details for this now, referencing the steps of the approach to analytical work earlier.

The key question then arises about how we aggregate them. This is an important problem since the user inputs the zoom level and which datasets to include. Hence a clear strategy is needed to cope with the heterogeneity of the number and which datasets are aggregated. Since most of our data is spatial raster data in the form of a TIFF we shall also make extensive use of the **rasterio** python package. And for the numerical calculations we will use **numpy and pandas.**

Data Selection

The first step is to select the data and extract relevant metrics that we wish to use to formulate the RDM index. The table below summarises the dataset, what part of the RDM profile it covers, its type and the relevant metrics that can be extracted from the dataset that can be used.

Dataset	RDM coverage	Type	Source	Access Type	Metrics	Polarity
Wind	clean energy	environmental	Global Wind Atlas	API (URL)	Mean Power Density	increasing
Solar	clean energy	environmental	Global Solar Atlas		Photovoltaic Power Potential	increasing
Pollution	environmental quality	environmental	World Air Quality Index	API (Request/Python)	AQI (Air Quality Index)	decreasing
Population Spatial Resolution	population	social	WorldPop	API (URL)	Population Density	decreasing

NDVI	agriculture	geographical		S3	NDVI	increasing

Data Loading/Imputation

Loading

Most of the data that we have selected is in GIS file in the TIF format - a particular raster method of storing geographical data. Each TIF file has a certain pixel height and width, with each pixel having a value which gives the metric value. To load tifs, we shall use the **rasterio** python package. The pixel data in the TIFs is usually loaded as a **numpy masked array**. We do this because a lot of countries' TIF files contain invalid, null values for pixels which are outside of the borders. Loading as a masked array enables us to directly access only valid values.

Imputation

For the data imputation, the main issue is combining all the data sources which each have a different geographic resolution. For example, the population density data is at 30arcseconds spatial resolution (about 1km at about equator), whereas the solar data is 9arcseconds. Thankfully, **rasterio** enables us to change the resolution of the TIFs by directly changing the height and width of the TIF file. For this, a **resampling method** (imputation/interpolation method) must be specified. We tested a variety of methods, but we found that a good compromise between speed and accuracy is the bilinear method. This is appropriate for non-linear and continuous data like what we have here for our RDM index heat map.

In order to get the TIFs in the same resolution, we need to set a standardised height and width for which they should conform to. One option is to take a user input here, however in theory there is not a guideline about what values to pick and lower resolutions will degrade the results, whilst higher resolutions will be numerically expensive. Another disadvantage here is that the RDM index would have to be recalculated completely every time a user resets the resolution. Instead, we set the population density resolution i.e. 30arcseconds as the standardised height and width. We have found the population density data to be most accurate of what we would expect in a country (where densely populated areas are) and also because it represents a lower bound on the spatial resolution.

Normalisation/Preparation

Scaling of Metrics

Then we must make sure the metrics are of comparable scale. This is an important first step as the metrics will originate from a wide range of datasets, each having their own intrinsic scale. It is also vital that we use a minimal number of mathematical transformations on the data. Otherwise, if we apply a unique transformation for each new metric the complexity of the scaling process increases indefinitely. All of the following are implemented using a combination of **numpy and pandas.**

We choose three different mathematical transformations to apply to the metrics: log_10 (base 10 logarithm), identity (do nothing) and Bi-symmetric Log transformation, defined by

$$\text{logbisymmetric}_b(x) = \text{sign}(x) * \log_b(1 + |x|)$$

Where b is the base of the logarithm. The purpose of adding this to our choice of transformations is that it is able to deal with metrics which also contain negative numbers and the value 0, as well as wide ranges of magnitudes. Note that for our purposes here we shall set b to be 10 to match the base of the logarithm of the first transformation. One can also change the 1 in the above to another constant if that is desired.

Since the number of included metrics could increase in the future, there should be a function that uses some criteria that automates the selection of the appropriate mathematical transformation to be applied to the inputted metric. Doing so avoids arbitrary decisions in which transformation to apply. The function should also be independent of the location since otherwise it would be impossible to compare different locations.

To decide which transformation to apply, we first define the span, in-span and out-span of a metric X.

$$\text{span}(X) = \max_{x \in X}(\log_{10}(|x|)) - \min_{x \in X}(\log_{10}(|x|))$$

$$\text{inspan}(X) \quad = \text{span}(x|x \in (X \cap (-1, 1))$$

$$\text{outspan}(X) \quad = \text{span}(x|x \in X \setminus [-1, 1]))$$

The first of these measures generally the range of magnitudes in X. The second measures the span of X in the interval between -1 and 1, whereas third measures the same but for X outside this interval. With this we can define the following pseudocode that is the algorithm which automatically decides between the transformations:

```
compute bothsigns(X) = whether X contains both numbers >0 and <0
compute haszeros(X) = whether X contains the value 0
compute inspan(X), outspan}(X)
if outspan(X) > maxspan:
      if (NOT haszeros(X)) AND (NOT bothsigns(X):
          Apply Log_10(|X|)
      else:
          Apply Logbisymmetric(X)
elif in_span > maxspan:
      Apply Logbisymmetric(X)
else:
      Apply identity(X)
```

The first if statement decides whether the outspan is bigger than some threshold **maxspan**. If it is, then the data is spread across many different magnitudes so we can consider the log and bi-symmetric log transformations. If the data does not contain zeros or positive and negative values at the same time, then we can apply the log, otherwise the bisymmetric log is more appropriate. If then the inspan is bigger than the same threshold then we apply the bisymmetric log because in this case there is a large order of magnitudes within the unit circle. This part is particularly relevant if we have a normalised index that is spread over orders of magnitudes e.g. the NDVI. If neither **out_span** or the **in_span** is true then the orders of magnitudes are within a reasonable range so we apply no transformation. After applying this algorithm, we z-score each metric.

Polarity

Once we have imputed the values so that they match up, we must make sure the polarity of the metric matches the aim of the RDM index. In particular, we want the maximum of the metric to correspond to a maximum in the RDM index. This is because certain increases in certain metrics may mean potential sites are more desirable, whilst the opposite could be true for other metrics. Note, that this does not even need to be a monotonic relationship. For example, we might like potential sites to have medium access to roads since high access would mean a high level of development is already present, and low access could become a barrier. In the above table, we have also added a column to represent the desired polarity for each metric. Those labelled '**increasing**' are the metrics which increase with increases in RDM index, and '**decreasing**' decrease with increases in RDM index. In the former case, no polarity transformation needs to be applied, but in the latter case we multiply the metric values by -1.

Applying Sigmoid function

Next, we need to make sure each of our metrics is in a bounded range. Not doing so means that the values of each metric after applying the scaling and polarity transformation is potentially infinite. A naive approach would be to normalise each metric by the maximum valid pixel value for each country. However, this raises a few major issues. Firstly, it would not make the method strictly speaking consistent and country independent since the maximum changes country to country. Secondly, if the maximum is actually an outlier you will normalise the index for that country by the wrong value. As an alternative, we use the commonly known sigmoid function defined as:

$$\sigma(X) = \frac{1}{1+e^{-X}}$$

Which takes an unbounded domain to the (0,1) interval. Note that from now on the $\sigma(X)$ will simply be denoted as X for convenience. $\sigma(X)$ has the advantage of providing one function regardless of the particular details of the country or metric. Hence we can be confident that comparing values across countries and metrics is valid. It is also not as sensitive to outliers compared to using the maximum as described earlier. Additionally, as we can see from the graph in the link it 'squashes' very high and very low values, making sure that there is a clearer distinction between desirable and undesirable sites.

Weighting/Aggregation

This part of the general approach essentially amounts to how we aggregate the metrics to find the RDM index. Below we provide these details. Since our data is generally static data, we do not build weights using conventional methods such as PCA/unsupervised learning since we do not have multiple samples of each metric. Our weighting/aggregation procedure is tailor made to fit our context. Again **numpy** makes these calculations very fast.

Calculating the RDM index

Here we describe how we calculate the RDM index. For a fixed location $i = 1, ..., N$ and category , we first calculate
$\tilde{X}_i^{(k)} = \mathrm{median}_j X_{ij}^{(k)}$,

Where $\tilde{X}_i^{(k)}$ is the k category median metric value at location i and where the median is taken over $j = 1, ..., M^{(k)}$, where the latter is the number of metrics for category k. $X_{ij}^{(k)}$ is the value of metric j at location i for category k. The rationale for calculating a separate aggregate metric for each category can be explained as follows. We can imagine a situation where if $M^{(k)}$ is much larger than some $M^{(k')}$ then aggregating all metrics together will ignore the important aspects coming from category k'. This biases the overall RDM index to that of category k. For similar reasons, we have chosen to use the median to aggregate metrics of the same category because this is a statistic that is more robust to any potential noise that might be more influential in categories with low sizes. The latter is particularly relevant since the number of metrics available at increasingly smaller zoom levels may be limited. Our strategy here, however, can adapt easily to the different number of datasets. Moreover, it gives added flexibility in case a user wants to see a composite index formed from just one category.

Now that we have an aggregated metric for each category, we now need to aggregate these further into the final RDM index. One option could be to simply aggregate the category medians by using the median again. However, another desirable feature would be the addition of user input for weights given to each category. For this, one would first need to calculate a default setting for the weights. To start, we first calculate the first version of the RDM index $RDM^{(1)}$ as

$$RDM_i^{(1)} = median_k \tilde{X}_i^{(k)} ,$$

Which is simply the median across all categories. In order to introduce a weighting scheme, we use the $RDM_i^{(1)}$ as a benchmark in the following way

$$w_k = 1 - \frac{\sum_i \left(\tilde{X}_i^{(k)} - RDM_i^{(1)}\right)^2}{\sum_{i,k'} \left(\tilde{X}_i^{(k')} - RDM_i^{(1)}\right)^2}$$

Where w_k is the weight associated with the category k. The numerator of the fraction represents the squared distance of the k category median to the overall median $RDM_i^{(1)}$. The corresponding denominator is a normalisation factor across all categories and locations. From this definition it has desirable properties, the main one being that as the category median becomes closer to $RDM_i^{(1)}$, w_k increases. Additionally, it is independent of the location i, which is also a vital property in order to fulfill one of the main aims of the RDM index to compare different locations. Finally we have

$$RDM_i^{(2)} = \sum_k w_k \tilde{X}_i^{(k)}$$

Where $RDM_i^{(2)}$ is the default weighted version of the RDM index.

Summary

The whole process detailed above can be summarised in a step by step process.

1. Check the polarity of the metric as detailed in that section.
2. Employ automatic scaling of the metrics using Scaling(Metric).
3. For selected datasets and their corresponding metrics, aggregate those in the same category (aggregate socials, economic, geographic etc) by finding $\tilde{X}_i^{(k)}$
4. If a composite version is required then calculate $RDM_i^{(1)}$ and $RDM_i^{(2)}$.
5. Overlay onto map.

Uncertainty/Sensitivity Analysis

We perform uncertainty analysis by adding white noise values to the values of the metric after it has been imputed and seeing how this affects the final results. As for sensitivity different metrics can be removed to see how the final results change.

Adding new metrics

The above framework offers a way to add metrics in the future that can be incorporated without having to change the framework’s underlying process given in the summary. Here is a step by step process when assessing whether a new metric can be added to the existing ones.

1. Check it adds information to the current list/with respect to the RDM profile. If it does, add the corresponding dataset to the data catalogue with all its attributes (name, description, update frequency, metric details etc).
2. Get some sample data from source (API, csv, web scraped etc).
3. Go through steps 1-5 in the summary.
4. See if any changes in results are desirable.