The Ocean Health Index: Modify a goal model using the Toolbox

Julia Stewart Lowndes Project Scientist, Ocean Health Index
lowndes@nceas.ucsb.edu

Introduction

The **Ocean Health Index (OHI)** measures the sustainable delivery of ten ‘goals’ (key benefits and services) to people now and in the future. At any spatial scale, OHI assessments use the best available information to develop goal models and set targets, and goal scores are combined together to give a comprehensive picture of coastal ocean health.

The OHI Toolbox is an open-source, cross-platform resource to calculate OHI scores at any scale. Here, we will use the Toolbox to add a new variable to a goal model. We will use Belize as an example: this is for teaching purposes only.

Note: this presentation is made with Rmarkdown. When rendered, it is viewable online and the code is available on GitHub.

Add a new variable to a goal model

We will walk through an example of adding a new variable to the Artisanal Fishing Opportunity (AO) goal model in Belize.

Assume your team has already discussed what is important in terms of AO in Belize. You have explored all data available and your team has decided that the best way to capture local characteristics is to add a new variable to the global AO model.

Add a new variable to a goal model

Your team has decided that the best way to capture local AO information is to add a new variable to the global AO model.

Steps:

1. Review global AO model
2. Prepare data layer & save in the `layers` folder
3. Register the layer in `layers.csv`
4. Update goal model in `functions.r`
5. Check and possibly update `goals.csv`
6. Calculate scores; view with WebApp

Step 1: Review AO global model

AO global model variables: need, access, and sustainability.

$$D_U = (1 - P) * (1 - O_{AO})$$
$$x_{AO} = (1 - D_U) * S_{AO}$$

D_U = unmet demand
P = need (PPP-adjusted per capita GDP)
O_{AO} = access to artisanal fishing
S_{AO} = sustainability of fishing methods

Step 1: Review AO global model

AO global model variables: need, access, and sustainability.

$$D_U = (1 - P) * (1 - O_{AO})$$
$$x_{AO} = (1 - D_U) * S_{AO}$$

The way this looks like in `functions.r`:

```
Du = (1.0 - need) * (1.0 - access)
statusData = ((1.0 - Du) * Sustainability)
```

Step 1: Plan AO global model

Let's say we have data available about poverty in Belize, which can be incorporated as part of need (the global model is based on need and access).

Step 1: Plan AO global model

Let's say we have data available about poverty in Belize, which can be incorporated as part of need (the global model is based on need and access).

Let's incorporate a second measure of need called P_{pov}.

$$D_U = (1 - (P + P_{pov})/2) * (1 - (O_{AO})$$

$$x_{AO} = (1 - D_U) * S_{AO}$$

Step 1: Plan AO global model

Let's say we have data available about poverty in Belize, which can be incorporated as part of need (the global model is based on need and access).

Let's incorporate a second measure of need called P_{pov}.

$$D_U = (1 - (P + P_{pov})/2) * (1 - (O_{AO})$$

$$x_{AO} = (1 - D_U) * S_{AO}$$

The way this looks like in `functions.r`:

```
Du = (1.0 - (need + need_pov) / 2 ) * (1.0 - access)
statusData = ((1.0 - Du) * Sustainability)
```

Step 2: Prepare & save new data layer

What do our new data look like?

- Population by district
- Percent below poverty line for Belize

Step 2: Prepare & save new data layer

What do our new data look like?

- Population by district
- Percent below poverty line for Belize

What do the existing data layers look like?

- **Explore using the blz WebApp**
  - `http://ohi-science.org/blz/layers`
- **Explore the blz repository**
  - access
  - need

Step 2: Prepare & save new data layer

Working with data: good practices

- **organize & prepare data in a github repository**
  - ex: `github/blz/blz-prep` (ours is stored here)

Working with data: good practices

- **organize & prepare data in a github repository**
  - ex: `github/blz/blz-prep` (ours is stored here)

- **document all metadata**
  - `README.md`s are readable on GitHub (example)
- **create a name-to-region lookup table**
  - ex: `rgn_district_lookup.csv`
- **script any data manipulations**
  - `prep_ao_need.r`

Step 2: Prepare & save new data layer

Our R script `prep_ao_need_pov.r` prepares the data layer called `ao_need_pov.csv` with these steps:

- Setup
- Access and clean raw data
- Calculate percent poverty by district
- Add region identifiers
- Save final data layer

Step 3. Register layer in `layers.csv`

Since we've created a new variable, we'll add a new row to `layers.csv`, with information for the following columns:

- **targets:** *AO*
- **layer:** `ao_need_pov`
- **name:** layer name will be displayed on your WebApp.
- **description:** description will be displayed on your WebApp.
- **fld_value:** percent
- **units:** percent
- **filename:** `ao_need_pov_blz2014.csv`
- **fld_id_num:** *rgn_id*

Step 4: Update model in `functions.r`

Update the model in `functions.r`, (and some layer handling):

```
Du = (1.0 - (need + need_pov)/2 ) * (1.0 - access)
statusData = ((1.0 - Du) * Sustainability)
```

Step 5: Check `goals.csv`

Look at `goals.csv` if there is anything to update (particularly in the *preindex_function* column)

Step 6: Calculate scores; view with WebApp

After you make your changes, you can calculate and visualize your scores in several ways:

1. In RStudio, run `calculate_scores.r` and visualize scores by running `launch_app_code.r`
2. In RStudio, run `calculate_scores.r`, sync changes to GitHub and visualize scores on the WebApp
3. Sync your changes to GitHub: scores will be calculated online and can be visualized on the WebApp