

Component Specification

The two main components of our project are (1) the Emissions Calculator and (2) the Dashboard Generator, with component interactions illustrated in Figure 1. The Emissions Calculator processes the data and outputs csv files that are used by the Dashboard Generator to visualize the processed data. Running the Emissions Calculator in advance allows the Dashboard Generator to quickly display the results, rather than re-calculating emissions impacts every time the website is loaded. We include the Emissions Calculator in our public repository so that a data analyst user could clone the repository, upload and specify new data, and re-run the Emissions Calculator for this updated data. This will be useful for future iterations of demand response (DR) projections, as the Northwest Power and Conservation Council continues to update projected DR potential and emissions rates for various policy scenarios.

Note: This design was inspired by a previous project for this class, `local_climate_change_tool`.

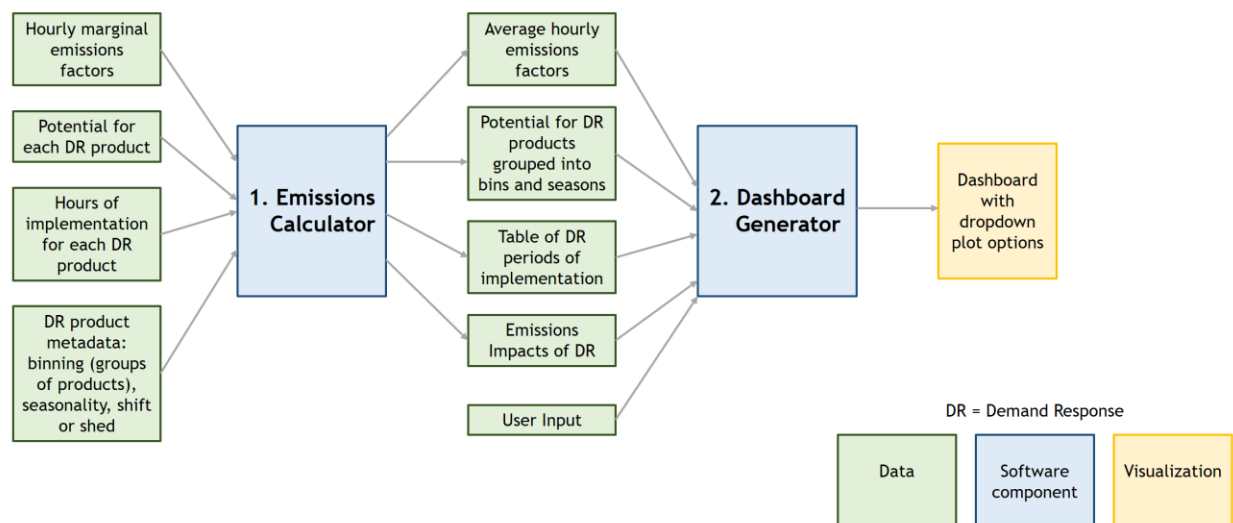


Figure 1. High-level component specification for the Northwest Demand Response Emissions Calculator project.

1. Emissions Calculator

What it Does:

- The primary goal of the emissions calculator is to calculate the emissions impacts of implementing demand response products in the Northwest U.S. To this end, the energy provided by DR products (MWh) is calculated by multiplying the projected potential for DR products (MW) by the hours of DR implementation (a 1 or 0 for each hour of the year, where 1 indicates DR

implementation). This emissions-free energy provided by DR replaces marginal fuel sources that would normally emit CO₂ at the marginal emissions rate (lbs CO₂e/kWh). The emissions savings due to implementing DR can therefore be calculated by multiplying the energy provided by DR (MWh) by the hourly marginal emissions rate (lbs CO₂e/kWh) and a units factor (1000 kWh/1MWh).

- For some DR products, the load is shifted rather than reduced; the emissions impacts therefore show the impact of shifting the load to the hours surrounding the DR implementation hours. If the emissions factors are higher during the hours of DR implementation compared to the adjacent hours, DR shift products will still reduce emissions. If the emissions factors are higher during the adjacent hours, DR shift products will increase emissions.
- The emissions calculator also outputs data to help visualize the components that produce these emissions impacts. This includes the average hourly emissions factors during days with DR implementation, the periods of DR implementation, and the DR product potential grouped by bin and season.
- *What are the DR bins?* Different DR products are grouped together in Bins 1 through 4, where Bin 1 products are more economically and technically feasible, and Bin 4 products are less feasible.
- *What are the DR plans?* The emissions calculator is set up to run for a list of DR plans, and a list of the seasons of implementation for each plan. Currently this includes newbins and oldbins. “oldbins” includes original DR projections provided by the Northwest Power and Conservation Council with 4 bins of DR products implemented in summer and winter, and “newbins” includes updated projections in which only two DR products, both in the new Bin 1, are implemented nearly every day during summer, fall, and winter.

Subcomponents A-D interact as part of the Emissions Calculator and are illustrated in Figure 2.

Required Inputs:

- Hourly marginal emissions rates (from excel file)
 - For each year, month, day, hour from 2022-2041
 - Hourly emissions rates at each timestep (lbs CO₂e/kWh; type: float)
 - Multiple excel files representing emissions factors for different policy scenarios can be specified, and the calculator will output results for each scenario
- DR hours (from excel file)
 - For each month, day, hour for one year, which will be applied to all years from 2022-2041
 - Each sheet of the excel file specifies the season
 - DR hours (type: integer; 1 or 0 every hour, with 1 indicating a DR hour) are specified for each DR product in different columns
- DR product potential (from excel file)
 - For each season, each year from 2022-2041, and each DR product

- DR potential (MW; type: float)
- DR product information (from excel file)
 - For each DR product, gives the bin, seasonality, shift or shed
 - These are all type: string
- The user will specify the DR plan names and seasons of implementation for each DR plan (type: lists of strings) within emissions_parameters.py
 - Multiple DR plans can be specified at once, each with different DR hours, potential, and product information, and the emissions calculator will run for all the specified plans.

For example:

```
DR_NAME = ['oldbins', 'newbins']
DR_SEASONS = [['Winter', 'Summer'], ['Winter', 'Summer', 'Fall']]
```

Outputs Provided:

- All outputs are csv files to be passed to the dashboard
- Average hourly marginal emissions rates:
 - Averaged over 2022-2041, only for days with DR implemented (DR hours includes a 1 on that day), for hour 1-24
 - One file for each DR plan, each season of implementation, and the average over all seasons of implementation
 - Using all days in 2022, one file for each seasonal and annual average
 - This will be shown in the default page for the general public to get a sense of typical emissions factors
- DR periods:
 - One file containing lists of strings for the periods of DR implementation for each DR plan and season, to be shown as a table
 - There will be one list of periods that apply to all direct load control (DLC) products and one list of periods that apply to all non-DLC products.
- DR potential:
 - One file for each DR plan, season, and bin of products, containing the annual DR potential for each product within that bin from 2022-2041
 - One comparison file containing the sum of DR potential across all products in a given bin for the last year of DR projections, here 2041, for each DR plan, season, and bin
- DR emissions impacts:
 - Time series and cumulative emissions impacts for 2022-2041 for each DR plan, each bin of products and all products combined, and each season and all seasons combined.
 - This includes different shifting options for residential time-of-use (resTOU): either assuming this is a shed product or shifting load to adjacent hours. While resTOU is typically specified as a shift product, some pilot studies suggest it may actually be a shed product.

Emissions Calculator Subcomponents:

A. Organize data:

Inputs: all required input excel files described above

Outputs: pandas dataframes and dictionaries of pandas dataframes to be used by the other subcomponents

B. Process emissions factors

Inputs: pandas dataframes for marginal emissions rates and DR hours

Outputs: pandas dataframes for average hourly marginal emissions rates

C. Calculate emissions impacts

Inputs: pandas dataframes for marginal emissions rates, DR hours, DR potential, DR product info

Outputs: pandas dataframes for DR emissions impacts

D. Output data

Inputs: pandas dataframes for DR hours, DR potential, average emissions rates, DR emissions impacts

Outputs: csv files containing the processed data for the dashboard

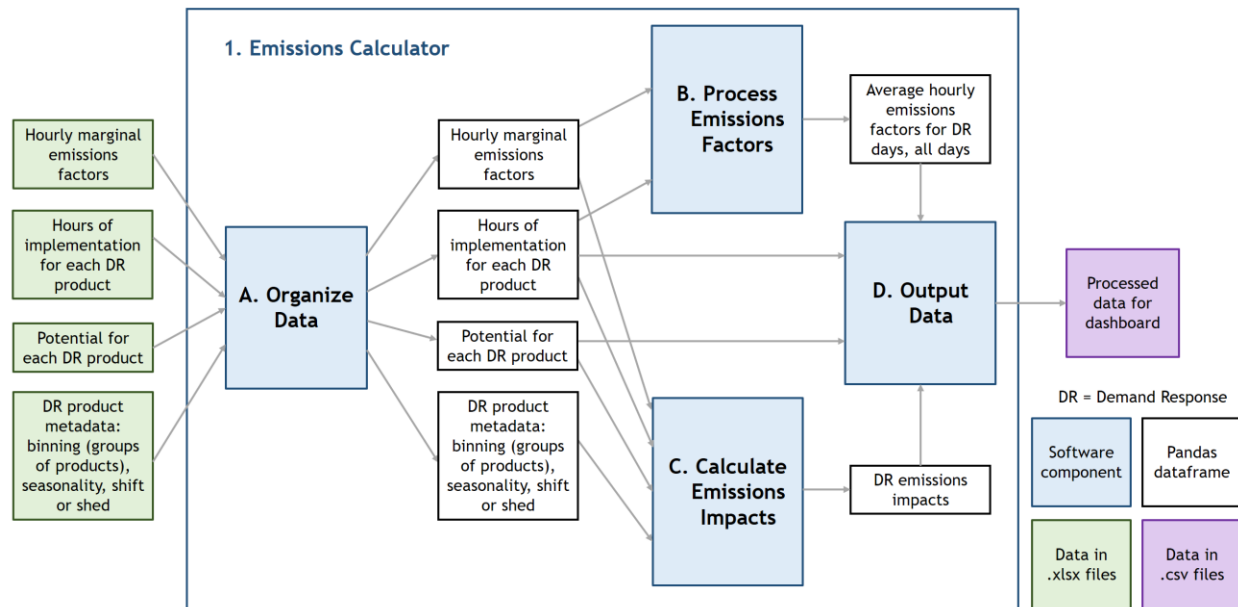


Figure 2. Subcomponent interactions schematic for component 1, the Emissions Calculator.

2. Dashboard Generator

What it Does:

The Dashboard creates graphics and displays them on an interactive website, where users can choose options from dropdowns to view different plots. This includes options for visualizing data for different DR plans, seasons, products or bins of products, and load shifting assumptions.

The “Home” page is geared toward a general public user, displaying typical daily emissions factors and the cumulative emissions impacts of DR. The “More Information” page is geared toward analysts and policymakers looking for more details, including the different components that produce the DR emissions impacts (DR potential, hours, emissions factors specific to DR days).

Subcomponents acquire data, generate visualizations, and update visualizations based on user input, as shown in Figure 3.

Required Inputs:

- Processed data from Emissions Calculator, all in csv files:
 - Average hourly emissions factors (float)
 - Potential for DR products grouped by bins and seasons (float)
 - DR periods of implementation (strings)
 - Emissions impacts of DR (float)
- User Input: choose whether to view the “Home” or “More Information” page; select dropdown options to view plots other than the default plots

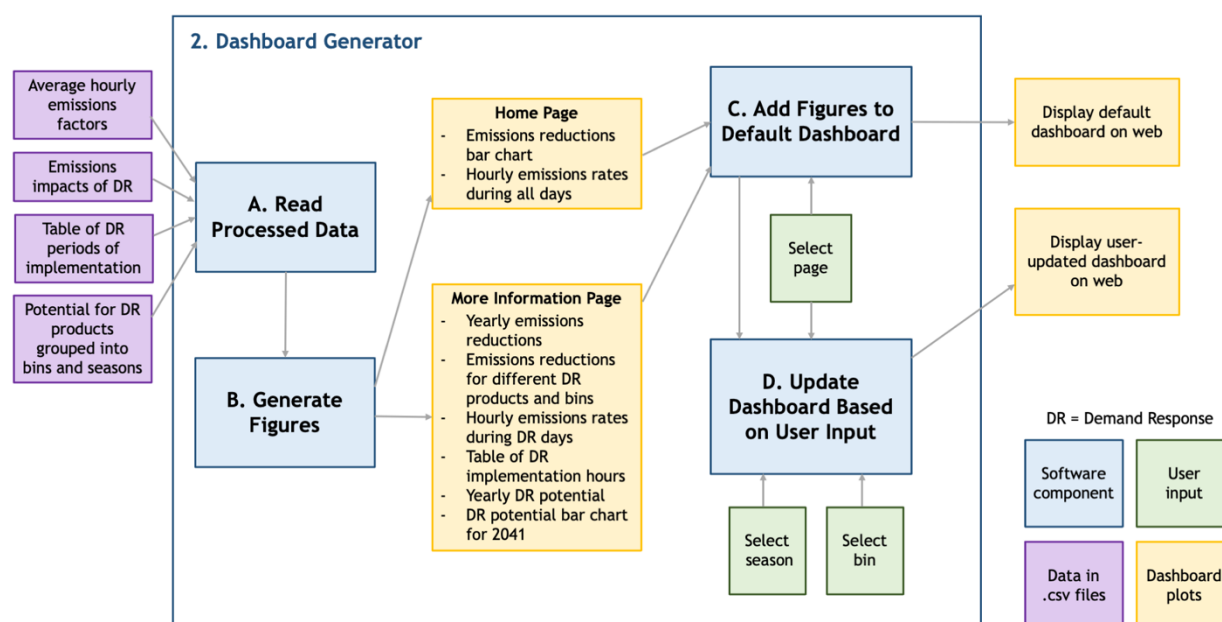


Figure 3. Subcomponent interactions schematic for component 2, the Dashboard Generator.

Outputs Provided:

Dashboard containing the following elements:

- A “Home” page and “More Information” page
- Plots for emissions factors, DR potential, emissions impacts
- Table with DR periods of implementation
- Dropdown options
- Interpretive descriptions

Dashboard Generator Subcomponents:

- A. Read in the data from the emissions calculator

Inputs: all of the csv files output by the emissions calculator

Outputs: pandas dataframes and dictionaries of pandas dataframes to be used for plotting, separated based on whether the data will be plotted on the “Home” or “More Information” page

- B. Make the figures

Inputs: pandas dataframes from reading in the csv files

Outputs: plotly graph objects for all of the plots and dropdown menus to be deployed on the web site

- C. Add figures to the default dashboard

Inputs: plotly graph objects for all of the plots and dropdown menus to be deployed on the dashboard

Outputs: default version of the dashboard, deployed on the web using Heroku

- D. Update the dashboard based on user input

Inputs: user choice of whether to see the “Home” or “More Information” page and which plots to show using the dropdown menus

Outputs: updated version of the dashboard

3. Use Case Interactions

Use Case 1

Objective:

The user is a data analyst who has received new emissions factors data for a new policy scenario. They would like to re-calculate emissions impacts of demand response using these new emissions factors and view detailed results.

Interactions:

The user first clones the repository, uploads the new emissions factors excel file, and updates emissions_parameters.py with the file information, all before running the Emissions Calculator. Within the Emissions Calculator, subcomponent 1A reads in all excel files, including the new emissions factor file, and outputs pandas dataframes to subcomponents 1B and 1C. Subcomponent 1B calculates average hourly emissions factors, and subcomponent 1C calculates DR emissions impacts. Subcomponent 1D uses this processed data outputted by 1B and 1C, in addition to DR hours and

potential from 1A, to output csv files for the Dashboard Generator. The Dashboard Generator reads in the data from the Emissions Calculator (subcomponent 2A), makes all of the figures and dropdown menus using plotly (subcomponent 2B), and adds all of the figures to the dashboard (subcomponent 2C). In order for the user to update the web site, the user must push all of the updated files and scripts to a GitHub repository, connect the repository to a Heroku account, and deploy the app to the web via Heroku. Once the dashboard is deployed, the user looks at the website and chooses alternate plot options, and the Dashboard Generator displays different plots (subcomponent 2D). The user downloads relevant plots to share with policymakers.

Use Case 2

Objective:

The user is a member of the general public who would like to understand how their electricity usage impacts fossil fuel emissions depending on the time of day, and would also like to learn about demand response and how it would impact emissions in the Northwest U.S.

Interactions:

The Emissions Calculator has already been run and has produced data for the Dashboard Generator, which is the only component with which the general public user will interact. Nevertheless, all of the graph objects are automatically made by the Dashboard Generator, so the user does not need to directly interact with any of the software subcomponents of the Dashboard Generator. The user loads the website and opens the “Home” page first. Displayed is a bar chart that shows the maximum total emissions reduction separated into different DR products, along with some interpretation. Also displayed is a plot of the annual hourly emissions factors; the user uses a dropdown to choose to view these factors for specific seasons or to compare all of the seasons in one plot. After reading some interpretation, the user chooses the “More Information” page. Displayed is a plot of the estimated yearly emissions reductions from 2022 to 2041, along with a dropdown to separate the plots by season. Also displayed is a bar chart that shows the maximum total emissions reduction for different DR products and binning strategies. Next, a plot of the hourly emissions rates during specifically DR days is displayed, along with a dropdown to compare seasons. Below is a table showing the time periods of DR implementation. Rounding out the “More Information” are plots of DR potential, with appropriate dropdowns to compare seasons and products.

4. Preliminary Plan

Phase 1: Draft the Emissions Calculator subcomponents while investigating the website

- a. emissions_parameters: Lily
- b. Subcomponent A: Lily - this should be done before starting B, C
- c. Subcomponent B: Chang
- d. Subcomponent C: James
- e. Subcomponent D:

- i. Lily - DR hours and potential from A
 - ii. Chang - Emissions factors from B
 - iii. James - Emissions impacts from C
- f. Dashboard:
 - i. Choose dashboard software: Daniel
 - ii. Figure out functionality using test data: Daniel
 - iii. Brainstorm what plots to show: Lily

Phase 1 should be completed by Monday Nov. 29.

Phase 2: Create website, iterate on Emissions Calculator, documentation

- a. Work on website subcomponents, add plots to website once output data is ready from Emissions Calculator: Daniel
- b. Update interpretive text for website: Lily
- c. Main script that runs through other scripts for emissions calculator: Lily
- d. Lily, Chang, James: give feedback on other subcomponents, add exceptions to own subcomponent, update tests, run pylint
- e. Other to-dos
 - i. Documentation
 - ii. Examples
 - iii. Continuous integration
 - iv. Setup.py

Phase 2 should create a website draft for project previews by Monday Dec. 6, a project preview presentation by Tuesday Dec. 7, and a final project and presentation by Wednesday Dec. 15.