User Guide Part 1: Emissions Calculator

Please follow the instructions below to install the emissions calculator, specify data inputs, and process the data to calculate emissions impacts of demand response.

Step 1: Clone the Repository

In your terminal, run the following commands to clone the repository and navigate to it:

```
git clone https://github.com/NW-Demand-Response-Emissions-
Impacts/emissions_calculator.git
cd emissions_calculator
```

Step 2: Set up your environment

Run the following commands:

```
conda create --name emissions_env
conda activate emissions_env
conda install pip
pip install -r requirements.txt
python setup.py install --user
```

Step 3: Upload new data

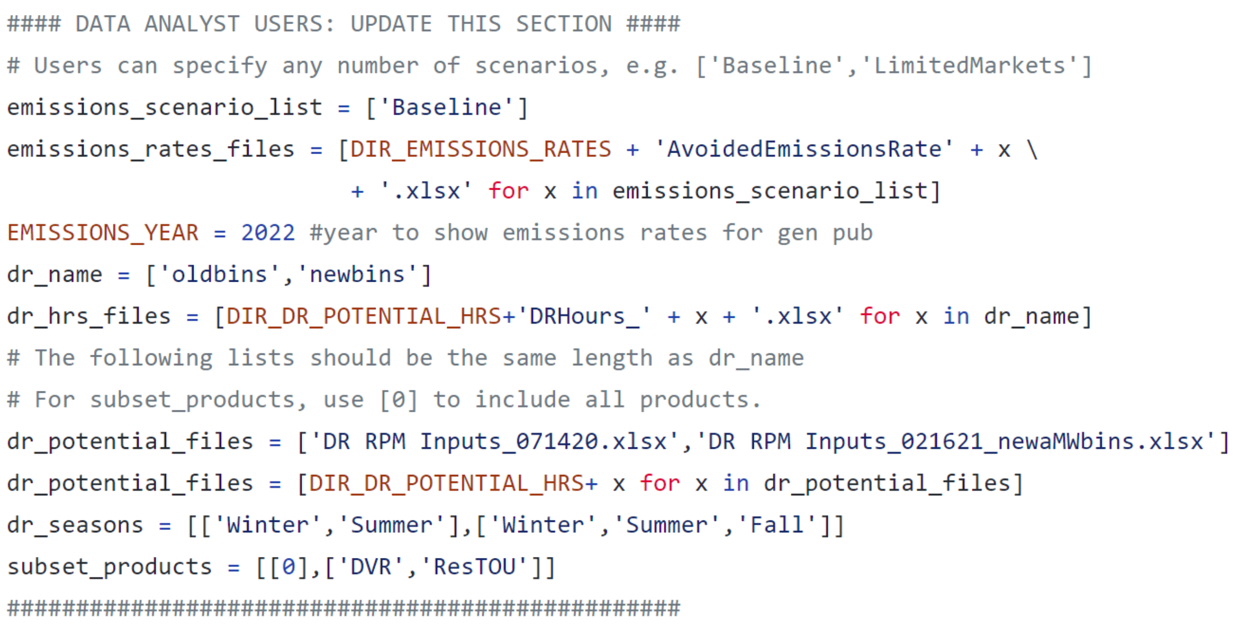
If you would like to use different data for the demand response potential and hours, add new excel files to the directory `data/input_data/DRPotentialandHours/`. To use different marginal emissions rates, add new excel files to the directory `data/input_data/AvoidedEmissionsRates/`.

Note that the emissions_calculator has been designed to run for excel files formatted in a particular way based on NW Power Council output data files, and will raise Value Errors if these formatting expectations are not met.

Step 4: Update data parameters and run the emissions calculator

1. Navigate to the `phase1_emissions_calculator/` directory by running: `cd emissions_calculator/phase1_emissions_calculator/`
2. Update data parameters in emissions_calculator.py within the section illustrated below:

######DATA ANALYST USERS: UPDATE THIS SECTION#####



```
#### DATA ANALYST USERS: UPDATE THIS SECTION ####
# Users can specify any number of scenarios, e.g. ['Baseline','LimitedMarkets']
emissions_scenario_list = ['Baseline']
emissions_rates_files = [DIR_EMISSIONS_RATES + 'AvoidedEmissionsRate' + x \
                         + '.xlsx' for x in emissions_scenario_list]
EMISSIONS_YEAR = 2022 #year to show emissions rates for gen pub
dr_name = ['oldbins','newbins']
dr_hrs_files = [DIR_DR_POTENTIAL_HRS+'DRHours_' + x + '.xlsx' for x in dr_name]
# The following lists should be the same length as dr_name
# For subset_products, use [0] to include all products.
dr_potential_files = ['DR RPM Inputs_071420.xlsx','DR RPM Inputs_021621_newaMWbins.xlsx']
dr_potential_files = [DIR_DR_POTENTIAL_HRS+ x for x in dr_potential_files]
dr_seasons = [['Winter','Summer'],['Winter','Summer','Fall']]
subset_products = [[0],['DVR','ResTOU']]
##################################################
```

This includes:

- emissions_scenario_list: a list of policy scenarios with emissions rates (the emissions calculator will determine impacts for each scenario)
- EMISSIONS_YEAR: a year for which the dashboard will display average emissions rates on the main page
- dr_name: a list of DR plan names
- dr_seasons: the seasons in which DR is implemented for each DR plan
- subset_products: the subset of DR products to consider for each DR plan
- emissions_rates_files, dr_potential_files, dr_hrs_files: file names for the marginal emissions rates, DR potential, and DR hours

3. Please note that some of the unit tests are specific to the default data parameters. Please update unit tests accordingly.
4. Run the emissions calculator by running: `python emissions_calculator.py`

Step 5: Update and interact with the dashboard

Please see the User Guide Part 2: Dashboard Generator for examples of how to update and interact with the dashboard.