Demand Response Emissions Impacts: Technology Review

Lily Hahn, Chang Liu, James Stadler, and Daniel Lloveras

16 November 2021

Background

- **The goal: create a dashboard to display the emissions impacts of demand response technologies in the NW**
  - Should be accessible by a non-technical audience, who don't use github or python → create a webpage display
  - Users include the general public (default page) and policymakers and analysts (more info page)

- **The dashboard should be able to:**
  - Access processed data from our github repository
  - Use dropdown options to show different plots
  - Create line plots, bar charts, pie charts from data; display schematic images and interpretation text; allow users to download plots

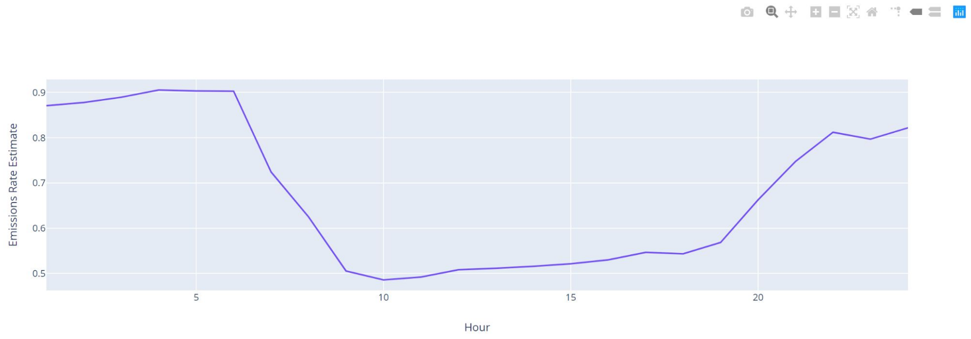
demand response: shifts or reduces electricity usage during periods of peak demand

Use case-Generate analysis of CO_2 emissions for a policymaker user

Interactions:

- User: Load dashboard webpage. (The emissions calculator has already processed the data and created files for the dashboard webpage in a github repository.)
- System: Show dashboard with a few default plots and interpretation; option to learn more
- User: Check out default plots; choose option to learn more
- System: Display new plots like emissions factors, dropdown option for different policy scenarios
- User: Choose alternate policy scenario for the emissions factor plot
- System: Show emissions factor plot for the alternate policy scenario
- User: Download plot

Average Hourly Avoided Emissions Rates for 2021-2041



Some descriptive text

Dropdown to choose another policy scenario or show lines for all scenarios

Learn more...

Package descriptions

Panel

- Interactive dashboard allows users to manipulate inputs and interact with graphs.
- Highly integratable with jupyter notebooks - Dashboard can be run within a notebook.
- Requires user to operate jupyter notebook.

Panel

Dash

- Similar interactive dashboard for python.
- Allows for a deployable dashboard accessible by url. Doesn't require any technical user skills.
- Built off of plotly, so requires use of plotly python packages.
- Can use HTML + CSS for customization

plotly | Dash

Comparison: Panel vs. Dash

Panel

- **Pros:**
  - Looks great in Jupyter Notebooks
  - Developer familiarity with matplotlib, Jupyter, hvPlot
- **Cons:**
  - *Jupyter Notebooks and GitHub are not accessible to non-technical users*
  - No need for fancy geospatial plotting

WINNER

plotly | Dash

- **Pros:**
  - *Easy for users: no technical skill required to view/interact with data*
  - Easy for developers: dashboard easy to deploy on web with Heroku using python script
  - Can access processed data using pandas
  - Options for interactivity with line/bar/pie charts
  - Widely used (lots of documentation)
- **Cons:**
  - Developer unfamiliarity with plotly, html