**Distributed Solar Technoeconomic Agent Characteristics (dSTAC): Documentation**

This data file summarizes key techno-economic metrics used in the NREL dGen model for modeling adoption of distributed solar by representative residential, commercial, and industrial entities for each county in the continental United States. As described further below, many of the metrics are derived as summaries of outputs from dGen. Specifically, each county and sector in this file is summarized as a single agent that is the weighted average of 10 statistically-representative agents (weighed by the statistical frequency). The dGen simulation used to derive this dataset was conducted in 2018 using the NREL 2018 Standard Scenario Mid Case assumptions (https://www.nrel.gov/docs/fy19osti/71913.pdf).

**Fields**

**state\_abbr**

Two letter US state name abbreviation.

**state\_fips**

Two digit US state FIPS code.[[1]](#footnote-1)

**county\_fips**

One - three digit US county FIPS code.[[2]](#footnote-2)

**sector\_abbr**

Three letter sector abbreviation. *res*: residential, *ind*: industrial, *com*: commercial

**energy\_value\_us\_dollars\_per\_kwh**

Annual value of energy from the electricity generated by the PV system calculated as the average total first year utility bill savings from the PV system (in US dollars) divided by the annual energy production of the system (in kWh). Thus, this represents the average *value* of distributed solar generation, which could be substantially different than the average *cost* of electricity (i.e. a tariff with a demand charge). Agents in dGen select a PV system capacity that optimizes the net present value of the considered investment, which could include a null system capacity. This metric was only calculated for agents having an optimal PV system size greater than zero.

Within the dGen model, PV systems size is constrained by the minimum of either the modeled rooftop area or generating 100% of the customer’s annual electricity consumption. Agent load is informed by distributions in EIA’s RECS and CBECS datasets[[3]](#footnote-3), while roof areas reflect distributions in Lidar-based analysis produced by Gagnon *et al* (2016)[[4]](#footnote-4). Local utility rates drawn from the Utility Rate Database[[5]](#footnote-5) are then used to calculate the optimal PV size for reducing a customer’s monthly electric bill from the utility.

**percent\_customers\_with\_nonzero\_sys\_size**

Percentage of representative customers per agent for which the optimal PV system calculated by dGen was greater than zero.

**avg\_roof\_sqft\_total**

The average total roof area in square feet across all representative customers suitable for PV development. Cumulative developable roof area by county is set to fixed totals interpolated from Lidar data. See footnote 4 below for more information on the distribution from which roof sizes where assigned to agents.

**avg\_roof\_sqft\_customers\_with\_nonzero\_sys\_size**

The average total roof area in square feet suitable for PV development for representative customers with optimal PV sizes calculated by dGen to be greater than zero.

**hourly\_capacity\_factor**

Hourly solar generation (kWh) for agent’s selected PV tilt and azimuth, per kWDC, i.e. kWh/kW. Put another way, it is the expected generation profile for a 1 kWDC system. This data is derived from George *et al* (2007).[[6]](#footnote-6) Data is stored as integers and each array entry must be divided by the *hourly\_capacity\_factor\_scalar* value to render generation in the correct unit (kWh/kW).

**hourly\_capacity\_factor\_scalar**

The value which each entry in the *hourly\_capacity\_factor* must be divided to render percent of annual energy production values.

N.B.: Since *hourly\_capacity\_factor* is stored as a string of a Python-style list, i.e. “[1,2…,3]”, users may need perform additional data manipulation to access to numeric values. We offer two suggestions for the Python and R languages:

**Python**

import pandas as pd

import numpy as np

df = pd.read\_csv('US\_States\_PV\_Adoption\_Agents.csv')

# Access the 0th index array

hourly\_cf = np.array(eval(df['hourly\_capacity\_factor'][0])) / float(df['hourly\_capacity\_factor\_scalar'][0])

# Access all arrays

df['hourly\_cf'] = df.apply( lambda x: np.array(eval(x.hourly\_capacity\_factor))/ float(x.hourly\_capacity\_factor\_scalar), axis=1)

**R**

df <- read.csv('~/Desktop/US\_States\_PV\_Adoption\_Agents.csv')

# Access 1st index array

array\_1 = df$hourly\_capacity\_factor[1]

scalar\_1 = df$hourly\_capacity\_factor\_scalar[1]

hourly\_cf\_1 <- as.integer(strsplit(substring(array\_1,2,nchar(as.character(array\_1))-1),',')[[1]])/scalar\_1

# Access all arrays

convert\_function <- function(x){

 list(as.integer(strsplit(substring(x[1],2,nchar(as.character(x[1]))-1),',')[[1]])/as.numeric(x[2]))

}

cols <- c("hourly\_capacity\_factor", "hourly\_capacity\_factor\_scalar")

df$hourly\_cf <- apply(df[cols],1,convert\_function)

1. See <https://transition.fcc.gov/oet/info/maps/census/fips/fips.txt> [↑](#footnote-ref-1)
2. See <https://transition.fcc.gov/oet/info/maps/census/fips/fips.txt> [↑](#footnote-ref-2)
3. For RECS see <https://www.eia.gov/consumption/residential/> , CBECS see <https://www.eia.gov/consumption/commercial/> [↑](#footnote-ref-3)
4. See <https://www.nrel.gov/docs/fy16osti/65298.pdf> [↑](#footnote-ref-4)
5. See <https://openei.org/apps/USURDB/> [↑](#footnote-ref-5)
6. See <https://www.nrel.gov/docs/gen/fy09/44443.pdf>. Also, see Appendix 1 in <https://www.nrel.gov/docs/fy16osti/65231.pdf> for more dGen specific data processing details. [↑](#footnote-ref-6)