How to get CAISO Day Ahead Electricity Market Price Data

Version 1.0 - Stable, Full Functionality
Brian Woo-Shem
Last updated: 2021-07-28

Automatically get any number of days CAISO wholesale price data in Python.
Simply run this program with type of data and what dates as shown below, and it will generate a .csv with wholesale price data in a single column at 5 minute timesteps that can be plugged directly into UCEF and the optimizer script.

Outputs csv with file name in format: WholesaleRealTime_2019-12-21_2020-3-30.csv

New script getWholesaleCAISO.py
SYNTAX:

```
        getWholesaleCAISO.py <data_type> <year> <month> <day> <number_of_days> <optional_options>

        <data_type>  d for day ahead, r for real time
        <year>       four digit year of first day to get (eg. 2020)
        <month>      month of the first day to get (eg. January, Jan, or 1)
        <day>        day of month for the first day (eg. 21)
        <number_of_days>   how many days to get data for. 1 = only get the day specified
        <optional_options>  Additional parameters, optional
          -v      verbose: provide extra outputs for user readability
          -d      debug: provide detailed outputs for debugging
          -leap   include leap years in date calculation
          -s      Server Safe mode; adds delay to avoid crashing server but takes longer.
          -fast   fast unsafe: removes all time delays for fastest data acquisition. WARNING: you are responsible if this mode breaks something or violates the CAISO Terms of Use.
```

For licensing info, run

```
        getWholesaleCAISO.py -l
```

Required Modules: beautifulsoup4, urllib3, numpy, os_sys, lxml, requests

----

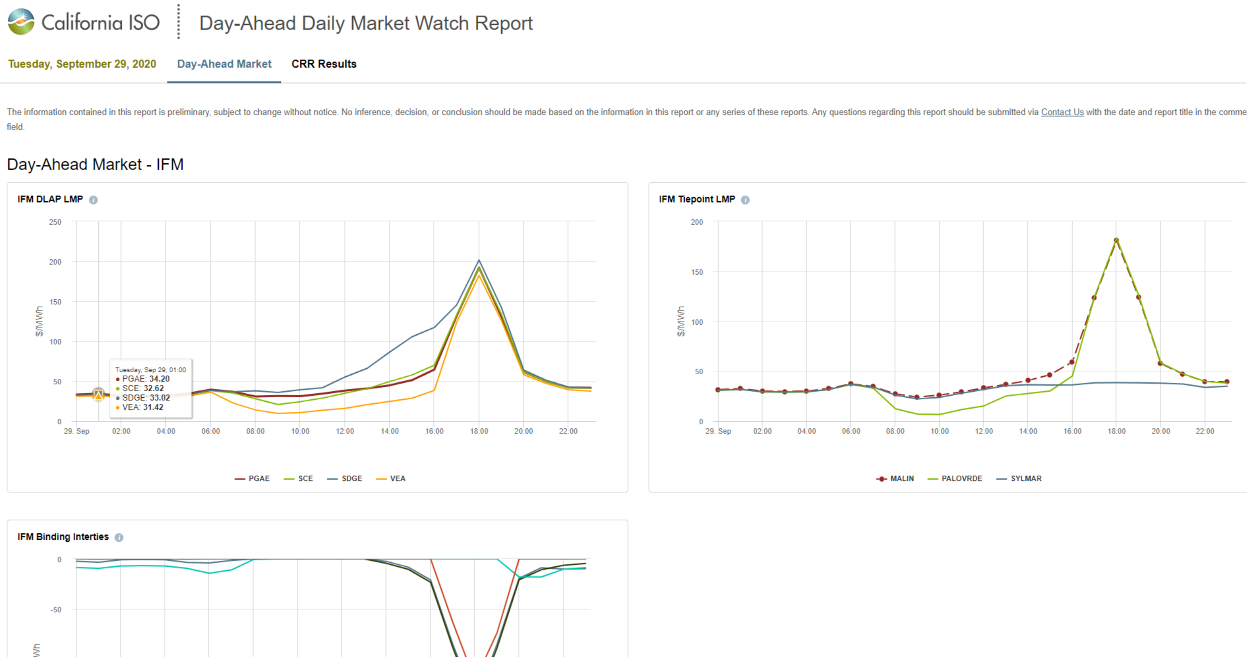
Version 0.9 - Stable, Partially automated only
2021-07-20

CAISO Day Ahead Market rates determine pricing in the optimization simulation. Simulation program requires data for every 5 minute timestep for the entire simulation in one long column, but the main data source only gives us a graph one day at a time. This process can quickly obtain and convert the data format.

Go to
www.caiso.com/Documents/Day-AheadDailyMarketWatchSep29-2020.html

Change the month, day, and year as indicated. For single digit days, use 01, 02,...

On the first graph, hover over so the PGAE line is selected. Notice that it provides the price by hour in the pop up box.



California ISO | Day-Ahead Daily Market Watch Report

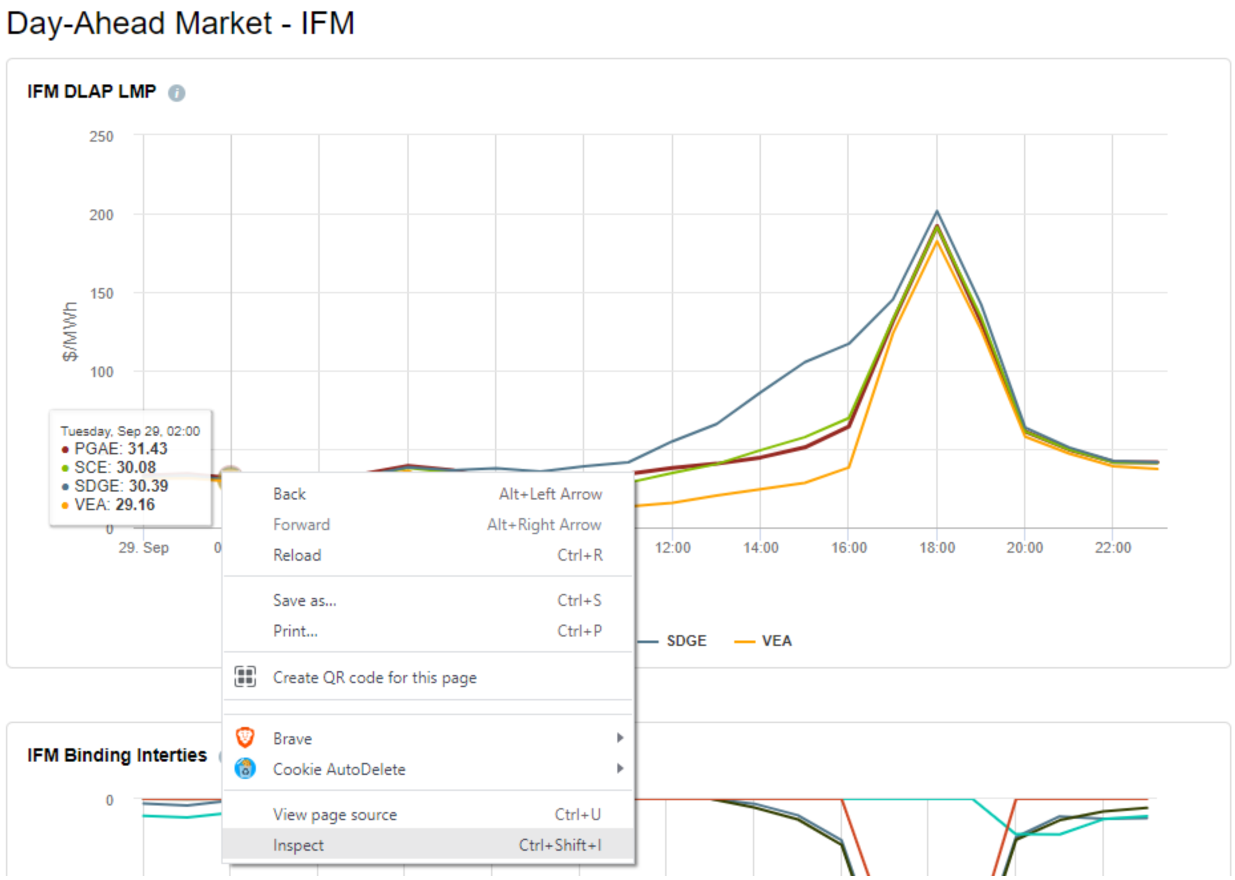
Tuesday, September 29, 2020 Day-Ahead Market CRR Results

The information contained in this report is preliminary, subject to change without notice. No inference, decision, or conclusion should be made based on the information in this report or any series of these reports. Any questions regarding this report should be submitted via Contact Us with the date and report title in the comme field.

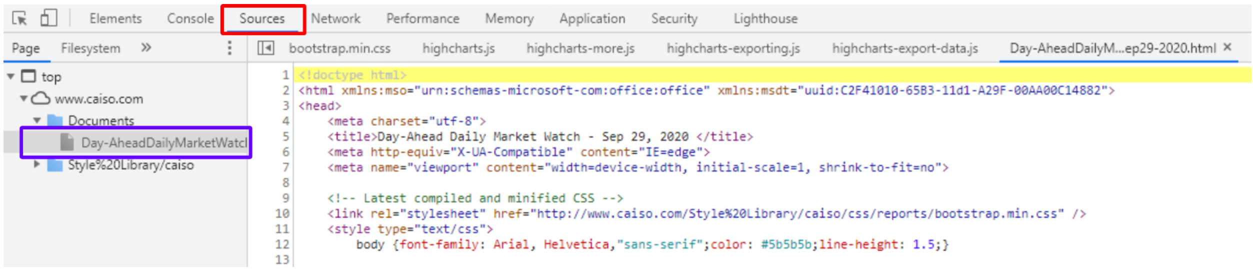
Day-Ahead Market - IFM

Tuesday, Sep 29, 01:00
● PGAE: 34.20
● SCE: 32.62
● SDGE: 33.02
● VEA: 31.42

Right click while still selecting this line, and click Inspect



In the Inspector Console window, go to the Sources tab and click Day-AheadDailyMarketWatch… on the left panel.

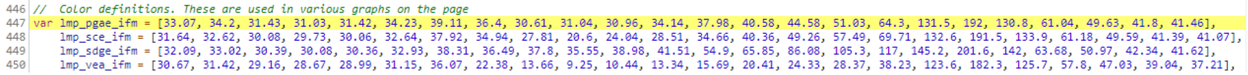


```
<!doctype html>
<html xmlns:mso="urn:schemas-microsoft-com:office:office" xmlns:msdt="uuid:C2F41010-65B3-11d1-A29F-00AA00C14882">
<head>
    <meta charset="utf-8">
    <title>Day-Ahead Daily Market Watch - Sep 29, 2020 </title>
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <!-- Latest compiled and minified CSS -->
    <link rel="stylesheet" href="http://www.caiso.com/Style%20Library/caiso/css/reports/bootstrap.min.css" />
    <style type="text/css">
        body {font-family: Arial, Helvetica,"sans-serif";color: #5b5b5b;line-height: 1.5;}

```

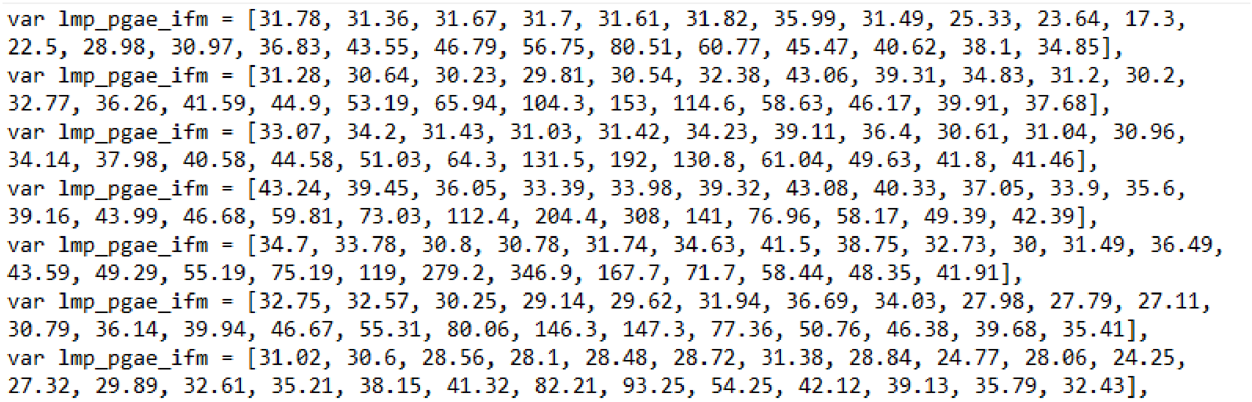
Scroll to line 447, it should say `var lmp_pgae_ifm` and then an array



```
446 //  Color definitions. These are used in various graphs on the page
447 var lmp_pgae_ifm = [33.07, 34.2, 31.43, 31.03, 31.42, 34.23, 39.11, 36.4, 30.61, 31.04, 30.96, 34.14, 37.98, 40.58, 44.58, 51.03, 64.3, 131.5, 192, 130.8, 61.04, 49.63, 41.8, 41.46],
448     lmp_sce_ifm = [31.64, 32.62, 30.08, 29.73, 30.06, 32.64, 37.92, 34.94, 27.81, 20.6, 24.04, 28.51, 34.66, 40.36, 49.26, 57.49, 69.71, 132.6, 191.5, 133.9, 61.18, 49.59, 41.39, 41.07],
449     lmp_sdge_ifm = [32.09, 33.02, 30.39, 30.08, 30.36, 32.93, 38.31, 36.49, 37.8, 35.55, 38.98, 41.51, 54.9, 65.85, 86.08, 105.3, 117, 145.2, 201.6, 142, 63.68, 50.97, 42.34, 41.62],
450     lmp_vea_ifm = [30.67, 31.42, 29.16, 28.67, 28.99, 31.15, 36.07, 22.38, 13.66, 9.25, 10.44, 13.34, 15.69, 20.41, 24.33, 28.37, 38.23, 123.6, 182.3, 125.7, 57.8, 47.03, 39.04, 37.21],
```

Copy this line into a text file.
Double check that the numbers match the PGAE line on the first graph.

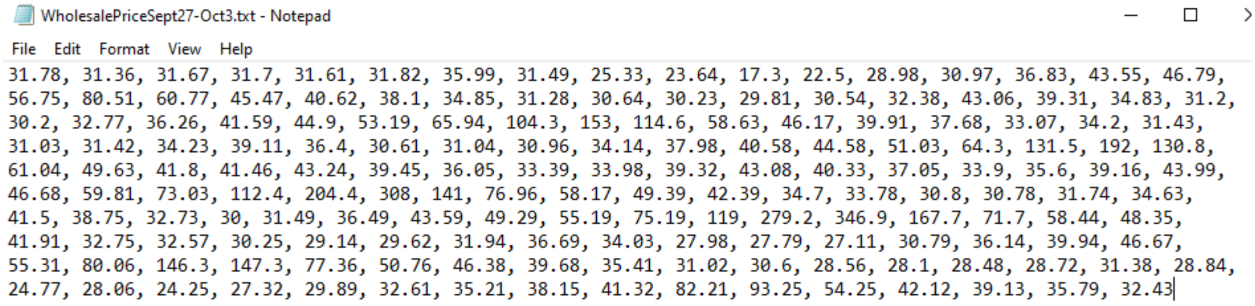
Repeat for as many days as needed. Once complete, the text file will look like this:



```
var lmp_pgae_ifm = [31.78, 31.36, 31.67, 31.7, 31.61, 31.82, 35.99, 31.49, 25.33, 23.64, 17.3,
22.5, 28.98, 30.97, 36.83, 43.55, 46.79, 56.75, 80.51, 60.77, 45.47, 40.62, 38.1, 34.85],
var lmp_pgae_ifm = [31.28, 30.64, 30.23, 29.81, 30.54, 32.38, 43.06, 39.31, 34.83, 31.2, 30.2,
32.77, 36.26, 41.59, 44.9, 53.19, 65.94, 104.3, 153, 114.6, 58.63, 46.17, 39.91, 37.68],
var lmp_pgae_ifm = [33.07, 34.2, 31.43, 31.03, 31.42, 34.23, 39.11, 36.4, 30.61, 31.04, 30.96,
34.14, 37.98, 40.58, 44.58, 51.03, 64.3, 131.5, 192, 130.8, 61.04, 49.63, 41.8, 41.46],
var lmp_pgae_ifm = [43.24, 39.45, 36.05, 33.39, 33.98, 39.32, 43.08, 40.33, 37.05, 33.9, 35.6,
39.16, 43.99, 46.68, 59.81, 73.03, 112.4, 204.4, 308, 141, 76.96, 58.17, 49.39, 42.39],
var lmp_pgae_ifm = [34.7, 33.78, 30.8, 30.78, 31.74, 34.63, 41.5, 38.75, 32.73, 30, 31.49, 36.49,
43.59, 49.29, 55.19, 75.19, 119, 279.2, 346.9, 167.7, 71.7, 58.44, 48.35, 41.91],
var lmp_pgae_ifm = [32.75, 32.57, 30.25, 29.14, 29.62, 31.94, 36.69, 34.03, 27.98, 27.79, 27.11,
30.79, 36.14, 39.94, 46.67, 55.31, 80.06, 146.3, 147.3, 77.36, 50.76, 46.38, 39.68, 35.41],
var lmp_pgae_ifm = [31.02, 30.6, 28.56, 28.1, 28.48, 28.72, 31.38, 28.84, 24.77, 28.06, 24.25,
27.32, 29.89, 32.61, 35.21, 38.15, 41.32, 82.21, 93.25, 54.25, 42.12, 39.13, 35.79, 32.43],
```

Remove all of the `var lmp_pgae_ifm = [` and `],` text. Get all the numbers separated by commas only. Use find and replace for more than a few days.

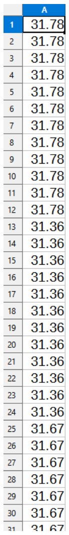
Should now look like this:



```
31.78, 31.36, 31.67, 31.7, 31.61, 31.82, 35.99, 31.49, 25.33, 23.64, 17.3, 22.5, 28.98, 30.97, 36.83, 43.55, 46.79,
56.75, 80.51, 60.77, 45.47, 40.62, 38.1, 34.85, 31.28, 30.64, 30.23, 29.81, 30.54, 32.38, 43.06, 39.31, 34.83, 31.2,
30.2, 32.77, 36.26, 41.59, 44.9, 53.19, 65.94, 104.3, 153, 114.6, 58.63, 46.17, 39.91, 37.68, 33.07, 34.2, 31.43,
31.03, 31.42, 34.23, 39.11, 36.4, 30.61, 31.04, 30.96, 34.14, 37.98, 40.58, 44.58, 51.03, 64.3, 131.5, 192, 130.8,
61.04, 49.63, 41.8, 41.46, 43.24, 39.45, 36.05, 33.39, 33.98, 39.32, 43.08, 40.33, 37.05, 33.9, 35.6, 39.16, 43.99,
46.68, 59.81, 73.03, 112.4, 204.4, 308, 141, 76.96, 58.17, 49.39, 42.39, 34.7, 33.78, 30.8, 30.78, 31.74, 34.63,
41.5, 38.75, 32.73, 30, 31.49, 36.49, 43.59, 49.29, 55.19, 75.19, 119, 279.2, 346.9, 167.7, 71.7, 58.44, 48.35,
41.91, 32.75, 32.57, 30.25, 29.14, 29.62, 31.94, 36.69, 34.03, 27.98, 27.79, 27.11, 30.79, 36.14, 39.94, 46.67,
55.31, 80.06, 146.3, 147.3, 77.36, 50.76, 46.38, 39.68, 35.41, 31.02, 30.6, 28.56, 28.1, 28.48, 28.72, 31.38, 28.84,
24.77, 28.06, 24.25, 27.32, 29.89, 32.61, 35.21, 38.15, 41.32, 82.21, 93.25, 54.25, 42.12, 39.13, 35.79, 32.43
```

Save the file as a .csv

Use the hourlyTo5min.py code to convert it to 5 minute data in columns needed by UCEF. Result looks like this:



	A
1	31.78
2	31.78
3	31.78
4	31.78
5	31.78
6	31.78
7	31.78
8	31.78
9	31.78
10	31.78
11	31.78
12	31.78
13	31.36
14	31.36
15	31.36
16	31.36
17	31.36
18	31.36
19	31.36
20	31.36
21	31.36
22	31.36
23	31.36
24	31.36
25	31.67
26	31.67
27	31.67
28	31.67
29	31.67
30	31.67
31	31.67