remake-insolation-data-measured

December 18, 2017

1 Extracting Insolation Data from the National Solar Radiation Data Base, 1991-2010 Update

The URL for this data base is http://rredc.nrel.gov/solar/old_data/nsrdb/1991-2010/. From there go to http://rredc.nrel.gov/solar/old_data/nsrdb/1991-2010/targzs/targzs_by_state.html for individual state and location data.

(See also: https://www.ncdc.noaa.gov/data-access/land-based-station-data/land-based-datasets/solar-radiation, but it seems dysfunctional.)

This follows on from the Nice paper and supports the Lyon IEEE paper.

File: remake-insolation-data-measured.ipynb. Copied from make-insolation-data-measured.ipynb, which was copied from make-Philly-insolation-data.ipynb, then edited for organizing the files. See README.txt.

```
In [35]: import pandas as pd
         import os
         import re
         # Include if needed:
         #%matplotlib
         # inline if needed
         #import matplotlib.pyplot as plt
```

1.1 Locate the Data

These places are on SOK's local machine, but hold data previously downloaded from the National Solar Radiation Data Base, 1991-2010 Update (see above).

Each location in the Solar Radiation Data Base is keyed with a unique ID number. (See above, find them at http://rredc.nrel.gov/solar/old_data/nsrdb/1991-2010/targzs/targzs_by_state.html for individual state and location data.)

Examples:

Pennsylvania:

PHL (Philadelphia International Airport): 724080

NE Philadelphia Airport (Wings Field): 724085

Allentown Lehigh Valley Intl Airport: 725170

Middletown Harrisburg Intl AP: 725115

New Jersey:

Atlantic City Intl AP: 724070

```
In [36]: # This program compiles insolation data for a single location,
         # called nsrdbLocation in the code, over the full 20 year period
         # covered in the data base.

         # Here we set the location to PHL, Philadelphia International Airport:
         nsrdbLocation = '725115'
         #'724070'
         #'725170'
         #'724085'
         #'724080'
         # Relevant for SOK only:
         #dataDir = '/Users/kimbrough/Dropbox/Processes/EnergyClimateChange/data/weathernoaanc
         #'/Users/stevenkimbrough/Dropbox/Processes/EnergyClimateChange/data/weathernoaancdc/n

         # Set dataDir, the directory we treat as the source of the data.
         dataDir = '../../Processes/EnergyClimateChange/data/nsrdb19912010/' + nsrdbLocation +
         # Set the output directory where we will write the CSV file, which will be moved late
         outputDir = '../../Processes/EnergyClimateChange/data/nsrdb19912010extracted/'
         # Get a list of the file names in the source directory.
         dataFiles = os.listdir(dataDir)
         # Filter the list of file names to remove any hidden files.
         dataFiles = [x for x in dataFiles if not x.startswith('.')]
         print(dataFiles)
```

```
['725115_1991_solar.csv', '725115_1992_solar.csv', '725115_1993_solar.csv', '725115_1994_solar
```

2 Important: Interpretation of Data Source

The User's manual for the Solar Radiation Data Base is at: http://www.nrel.gov/docs/fy12osti/54824.pdf

Here is what it says about 'METSTAT Glo (Wh/m^2)' (column P):

Total amount of direct and diffuse solar radiation (MATSTATE-modeled) received on a horizontal surface during the 60-minute period ending at the timestamp

Here is what it says about 'METSTAT Dir (Wh/m^2)' (column R):

Amount of solar radiation (METSTAT-modeled) received in a collimated beam on a surface normal to the sun during the 60-minute period ending at the timestamp

In general, 'METSTAT Dir (Wh/m^2)' will be larger than 'METSTAT Glo (Wh/m^2)' because it is optimally positioned towards the sun for the time. This is close to what we would expect tracking solar panels to achieve. Prudently placed fixed solar panels would be oriented much closer to normal to the sun than horizontal panels, so 'METSTAT Glo (Wh/m^2)' is too low for realistic installations.

We would extract 'METSTAT Dir (Wh/m^2)' and use that as a default, but some of the data seems flakey, e.g., 1996. In the future it could be modified to better approximate fixed panels.

So, we end up with the under-estimate, 'METSTAT Glo (Wh/m^2)'.

```
In [37]: columnName = 'METSTAT Glo (Wh/m^2)'
         columnFileName =  '-METSTAT_Glo_(Whm2)' #'-METSTAT Dir (Whm2)' # valid in a file name
```

```
        # Extract the data column for the first file.
        # Read in the first file.
        DFFull = pd.read_csv(dataDir + os.sep + dataFiles[0])
        print(DFFull.columns)
        insolationSeries = DFFull[columnName].values # 'METSTAT Glo (Wh/m^2)'
        # We take only the first 8760 hours of the year, thereby excluding
        # The final day of leap years.
        df = pd.DataFrame(insolationSeries[:8760],columns=['1991'])
        #For measured data: Meas Glo (Wh/m^2)
        df.head(12)
        # Good, this works.
        # So now we have df for 1991. It remains to read in the rest of the files
        # and extract their 'METSTAT Dir (Wh/m^2)' columns into df.
```

```
Index(['YYYY-MM-DD', 'HH:MM (LST)', 'Zenith (deg)', 'Azimuth (deg)',
       'ETR (Wh/m^2)', 'ETRN (Wh/m^2)', 'SUNY Glo (Wh/m^2)', 'SUNY Glo Flg',
       'SUNY Glo Unc (%)', 'SUNY Dir (Wh/m^2)', 'SUNY Dir Flg',
       'SUNY Dir Unc (%)', 'SUNY Dif (Wh/m^2)', 'SUNY Dif Flg',
       'SUNY Dif Unc (%)', 'METSTAT Glo (Wh/m^2)', 'METSTAT Glo Unc (%)',
       'METSTAT Dir (Wh/m^2)', 'METSTAT Dir Unc (%)', 'METSTAT Dif (Wh/m^2)',
       'METSTAT Dif Unc (%)', 'METSTAT CSKY Glo (Wh/m^2)',
       'METSTAT CSKY Glo Unc (%)', 'METSTAT CSKY Dir (Wh/m^2)',
       'METSTAT CSKY Dir Unc (%)', 'METSTAT CSKY Dif (Wh/m^2)',
       'METSTAT CSKY Dif Unc (%)', 'Meas Glo (Wh/m^2)', 'Meas Glo Flg',
       'Meas Dir (Wh/m^2)', 'Meas Dir Flg', 'Meas Dif (Wh/m^2)',
       'Meas Dif Flg', 'Precip Wat (cm)', 'Precip Wat Flg', 'AOD (unitless)',
       'AOD Flg', 'AOD RAN (unitless)', ' AOD RAN Flg', 'Ozone (cm)',
       'Ozone Flg', 'Albedo (unitless)', 'Albedo Flg'],
      dtype='object')
```

```
Out[37]:     1991
        0      0
        1      0
        2      0
        3      0
        4      0
        5      0
        6      0
        7      7
        8    108
        9    253
        10   371
        11   439
```

```
In [38]: df.tail(12)
```

```
Out[38]:      1991
        8748   371
```

```
8749    323
8750    226
8751    109
8752     18
8753      0
8754      0
8755      0
8756      0
8757      0
8758      0
8759      0
```

```
In [39]: # Now do the rest of the years
         year = 1991

         for file in range(1,len(dataFiles)):
             DFFull = pd.read_csv(dataDir + os.sep + dataFiles[file])
             year += 1
             #print(len(DFFull['METSTAT Dir (Wh/m^2)'].values))
             # This lops off the extra day during leap years.
             df[str(year)] = DFFull[columnName].values[:8760]
```

```
In [40]: df.shape
```

```
Out[40]: (8760, 20)
```

```
In [41]: df.head(12)
```

```
Out[41]:     1991  1992  1993  1994  1995  1996  1997  1998  1999  2000  2001  2002  \
         0      0     0     0     0     0     0     0     0     0     0     0     0
         1      0     0     0     0     0     0     0     0     0     0     0     0
         2      0     0     0     0     0     0     0     0     0     0     0     0
         3      0     0     0     0     0     0     0     0     0     0     0     0
         4      0     0     0     0     0     0     0     0     0     0     0     0
         5      0     0     0     0     0     0     0     0     0     0     0     0
         6      0     0     0     0     0     0     0     0     0     0     0     0
         7      7     5     5     7     2     3     5     6     4     6     8     7
         8    108    83    48    96    24    39    36   100    81    97   107   108
         9    253   210    70   206    45    83    79   222   237   234   253   253
         10   371   323   171   333    97   117   113   196   238   341   358   372
         11   439   322   267   312    81   136   133   124   366   415   442   442

             2003  2004  2005  2006  2007  2008  2009  2010
         0      0     0     0     0     0     0     0     0
         1      0     0     0     0     0     0     0     0
         2      0     0     0     0     0     0     0     0
```

```
3      0    0    0    0    0    0    0    0
4      0    0    0    0    0    0    0    0
5      0    0    0    0    0    0    0    0
6      0    0    0    0    0    0    0    0
7      3    8    9    3    7    3    8    4
8     34  109  113   34   65   84  112   74
9     53  253  257   66  232   98  260  214
10    92  370  374   99  346  142  379  321
11   108  438  441  118  407  439  448  175
```

```
In [42]: df.tail(12)
```

```
Out[42]:       1991  1992  1993  1994  1995  1996  1997  1998  1999  2000  2001  2002  \
         8748   371   112   135   138   401   330   388   139   405   430   438   370
         8749   323    99   121   124   184   354   346   316   384   374   389   380
         8750   226    51    92    95   114   111   222    96   282   286   287   280
         8751   109    35    52    54    64    56   134    55   164    72   149    66
         8752    18    11    10    10    12    26    27    29    31    15    24    17
         8753     0     0     0     0     0     0     0     0     0     0     0     0
         8754     0     0     0     0     0     0     0     0     0     0     0     0
         8755     0     0     0     0     0     0     0     0     0     0     0     0
         8756     0     0     0     0     0     0     0     0     0     0     0     0
         8757     0     0     0     0     0     0     0     0     0     0     0     0
         8758     0     0     0     0     0     0     0     0     0     0     0     0
         8759     0     0     0     0     0     0     0     0     0     0     0     0

               2003  2004  2005  2006  2007  2008  2009  2010
         8748   453   140   135   131   434   301    97    95
         8749   405   126   101   117   385   337    85   345
         8750   304    97    77    88   282   262    62    65
         8751   165    56    43    48   146   153    37    37
         8752    31    11     8    11    24    28    13    19
         8753     0     0     0     0     0     0     0     0
         8754     0     0     0     0     0     0     0     0
         8755     0     0     0     0     0     0     0     0
         8756     0     0     0     0     0     0     0     0
         8757     0     0     0     0     0     0     0     0
         8758     0     0     0     0     0     0     0     0
         8759     0     0     0     0     0     0     0     0
```

2.0.1 Here:

```
In [43]: # Write df to a CSV and we are done.
         df.to_csv(outputDir + nsrdbLocation + columnFileName + '-nsrdb.csv')
```

```
In [44]: #!jupyter nbconvert --to pdf remake-insolation-data-measured.ipynb
```

```
In [ ]:
```