

emmaredfoot / RAVEN\_APS

Branch: master ▾ RAVEN\_APS / eia\_datetime\_general.py

Find file Copy path

emmaredfoot Python document which organizes eia data

9b17cab a minute ago

1 contributor

56 lines (52 sloc) | 2.98 KB

```

1  #Code to break the EIA APS demand data into separate seasons
2  import csv
3  import os
4  #Create a csv file to write the data
5  def writeFile(outfile, monthB, monthE, dayB, dayE):
6      #Open the Arizona Public Service Demand data downloaded from EIA
7      dataset = csv.reader(open('RAVEN/APS_data_1.csv', newline=''), delimiter=',')
8      with open(outfile, 'w') as subfile:
9          # Rewrite the first column in the eia data so that it gives the year, the month and day, and the times
10         subfile.writelines('Time,Demand'+os.linesep)
11         for row in dataset:
12             day = row[0][8:10]
13             month = row[0][5:7]
14             hour = row[0][-5:-3]
15             demand = row[1]
16             #First check if the months correspond to those in the given season
17             if monthB <= int(month)<= monthE:
18                 #If the month is the last month, make sure that it is only those days on or before the beginning
19                 if int(month) == monthE:
20                     if int(day) <= dayE:
21                         subfile.writelines('{0},{1}'.format(row[0][-5:-3], row[1]+os.linesep))
22                 elif monthB < int(month) < monthE:
23                     subfile.writelines('{0},{1}'.format(row[0][-5:-3], row[1]+os.linesep))
24                 #If the month is the first month, only write those days on or after the final day
25                 elif int(day) >= dayB:
26                     subfile.writelines('{0},{1}'.format(row[0][-5:-3], row[1]+os.linesep))
27
28 def writeWinter(outfile, monthB, monthE, dayB, dayE):
29     #Open the Arizona Public Service Demand data downloaded from EIA
30     dataset = csv.reader(open('RAVEN/APS_data_1.csv', newline=''), delimiter=',')
31     with open(outfile, 'w') as subfile:
32         # Rewrite the first column in the eia data so that it gives the year, the month and day, and the times
33         subfile.writelines('Time,Demand'+os.linesep)
34         for row in dataset:
35             day = row[0][8:10]
36             month = row[0][5:7]
37             hour = row[0][-5:-3]
38             demand = row[1]
39             #First check if the months correspond to those in the given season
40             if monthB <= int(month)+12 <= monthE+12:
41                 #If the month is the last month, make sure that it is only those days on or before the beginning
42                 if int(month) == monthE:
43                     if int(day) <= dayE:
44                         subfile.writelines('{0},{1}'.format(row[0][-5:-3], row[1]+os.linesep))
45                 elif monthB < int(month)+12 < monthE+12:
46                     subfile.writelines('{0},{1}'.format(row[0][-5:-3], row[1]+os.linesep))
47                 #If the month is the first month, only write those days on or after the final day
48                 elif int(day) >= dayB:
49                     subfile.writelines('{0},{1}'.format(row[0][-5:-3], row[1]+os.linesep))
50
51
52 Spring = writeFile('RAVEN/SpringEIA.csv', 3, 6, 20, 20)
53 Summer = writeFile('RAVEN/SummerEIA.csv', 6, 9, 22, 21)
54 Fall = writeFile('RAVEN/FallEIA.csv', 9, 12, 22, 21)
55 Winter = writeWinter('RAVEN/WinterEIA.csv', 12, 3, 21, 19)

```

