ene-wind-solar-summary

October 2, 2019

```
In [33]: import sqlite3
         import pandas as pd
         import matplotlib.pyplot as plt
         from pandasql import sqldf
         sdf = lambda q: sqldf(q, globals())

         # Create the connection to the unzipped database
         cnx  = sqlite3.connect(r'./data-oasis-ene-wind-solar-summary_00.db')
```

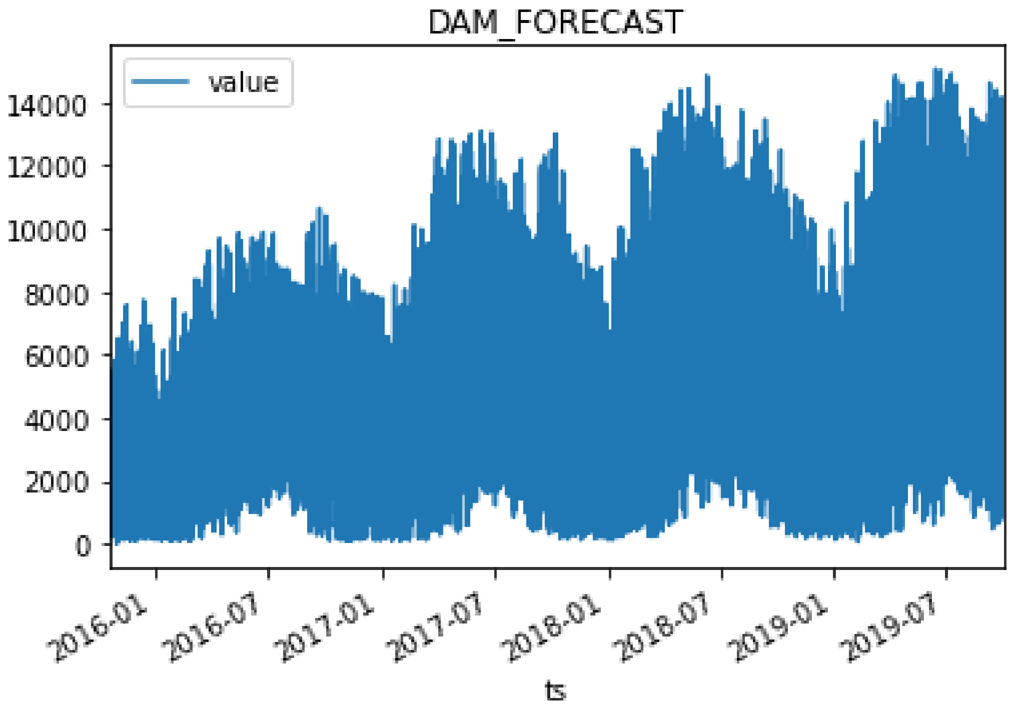
```
In [34]: df = pd.read_sql("select report_data.data_item, report_data.value, report_header.uom,
         df['ts'] = pd.to_datetime(df['interval_start_gmt'])
         df = df.set_index('ts')
         df.drop(['interval_start_gmt'], axis=1, inplace=True)
         df[0:5]
```

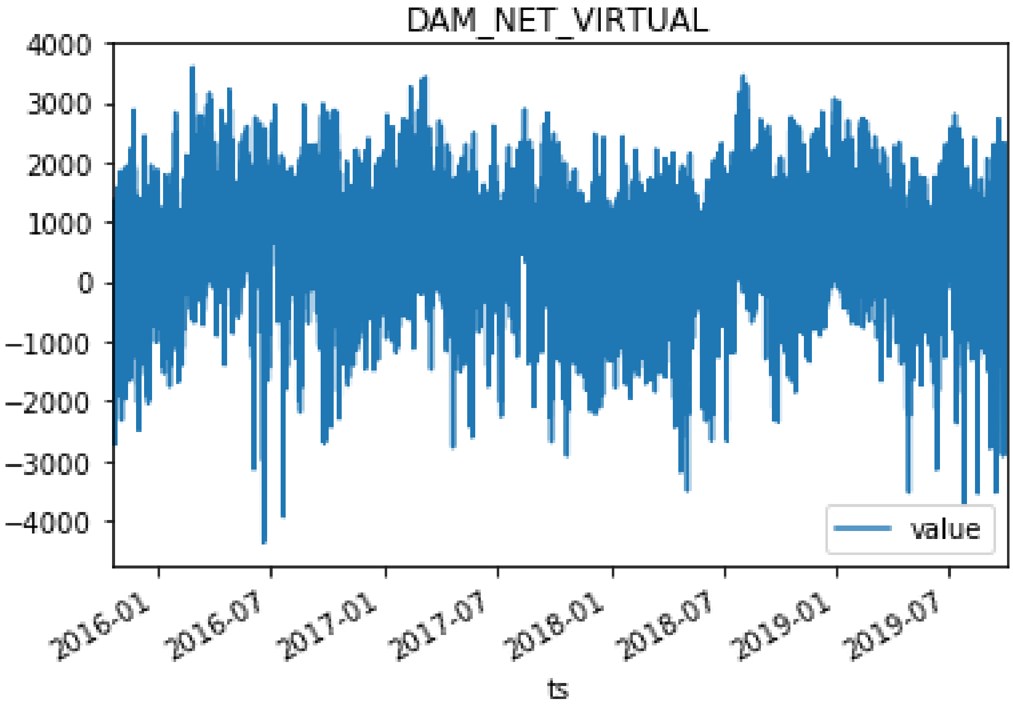
```
Out[34]:                                  data_item   value uom  \
         ts
         2016-11-15 07:00:00+00:00     DAM_FORECAST  364.00  MW
         2016-11-15 07:00:00+00:00  DAM_NET_VIRTUAL  453.00  MW
         2016-11-15 07:00:00+00:00     DAM_SCHEDULE  173.00  MW
         2016-11-15 07:00:00+00:00     RTM_SCHEDULE  311.42  MW
         2016-11-15 08:00:00+00:00     DAM_FORECAST  612.00  MW

                                                  report mkt_type  \
         ts
         2016-11-15 07:00:00+00:00  ENE_WIND_SOLAR_SUMMARY      DAM
         2016-11-15 07:00:00+00:00  ENE_WIND_SOLAR_SUMMARY      DAM
         2016-11-15 07:00:00+00:00  ENE_WIND_SOLAR_SUMMARY      DAM
         2016-11-15 07:00:00+00:00  ENE_WIND_SOLAR_SUMMARY      RTD
         2016-11-15 08:00:00+00:00  ENE_WIND_SOLAR_SUMMARY      DAM

                                          interval_end_gmt
         ts
         2016-11-15 07:00:00+00:00  2016-11-15T08:00:00-00:00
         2016-11-15 07:00:00+00:00  2016-11-15T08:00:00-00:00
         2016-11-15 07:00:00+00:00  2016-11-15T08:00:00-00:00
         2016-11-15 07:00:00+00:00  2016-11-15T08:00:00-00:00
         2016-11-15 08:00:00+00:00  2016-11-15T09:00:00-00:00
```

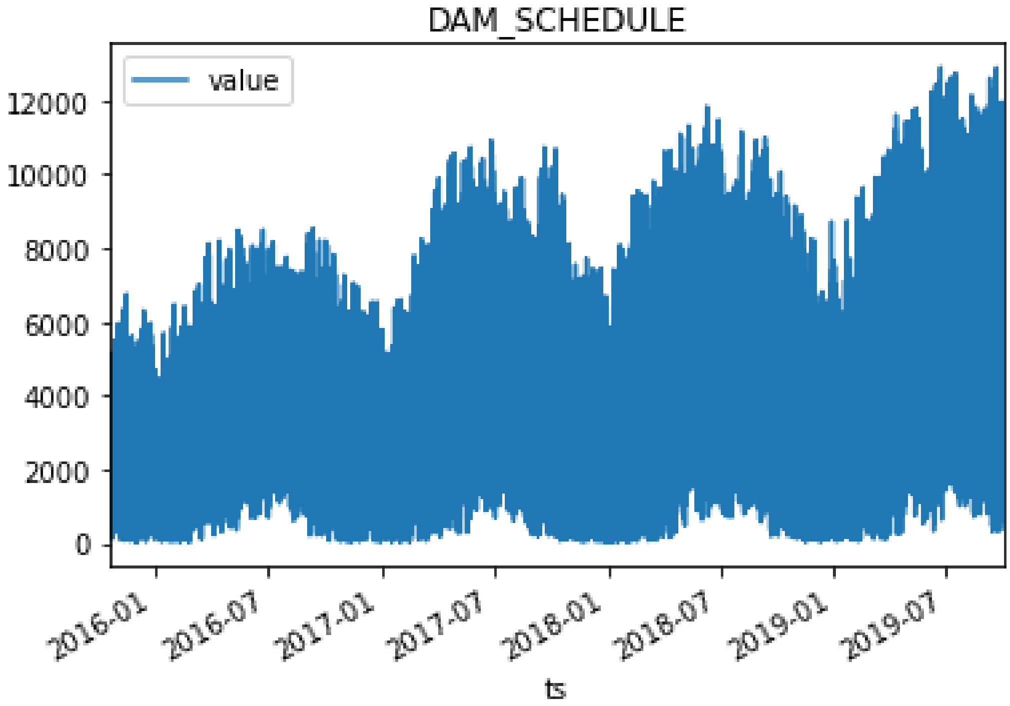
```
In [35]: df.info()
```

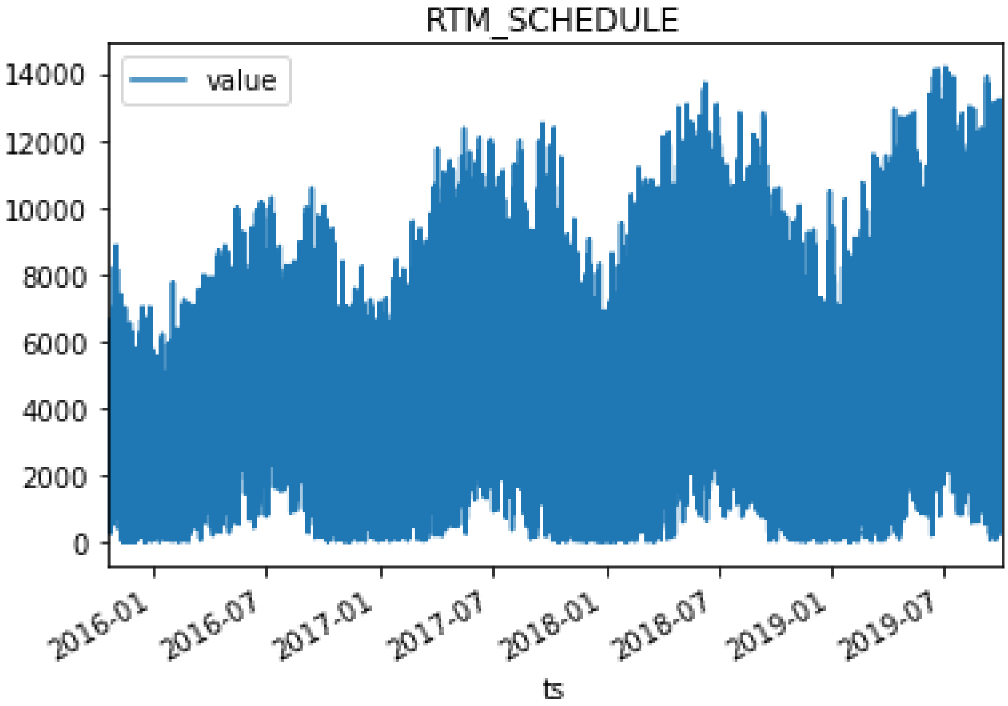
```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 138354 entries, 2016-11-15 07:00:00+00:00 to 2019-10-01 06:00:00+00:00
Data columns (total 6 columns):
data_item           138354 non-null object
value               138354 non-null float64
uom                 138354 non-null object
report              138354 non-null object
mkt_type            138354 non-null object
interval_end_gmt    138354 non-null object
dtypes: float64(1), object(5)
memory usage: 7.4+ MB
```

```
In [36]: for title, group in df.groupby(['data_item']):
             group.plot.line(y='value', title=title)
```



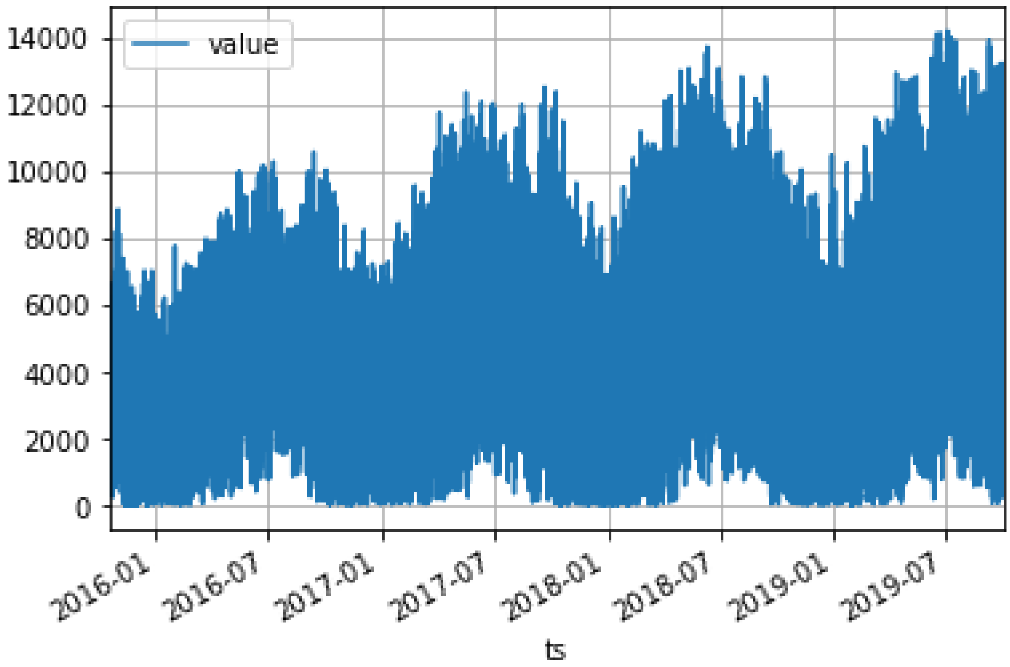

DAM_NET_VIRTUAL
4000
3000
2000
1000
0
−1000
−2000
−3000
−4000
value
2016-01
2016-07
2017-01
2017-07
2018-01
2018-07
2019-01
2019-07
ts


DAM_SCHEDULE
value
12000
10000
8000
6000
4000
2000
0
2016-01
2016-07
2017-01
2017-07
2018-01
2018-07
2019-01
2019-07
ts

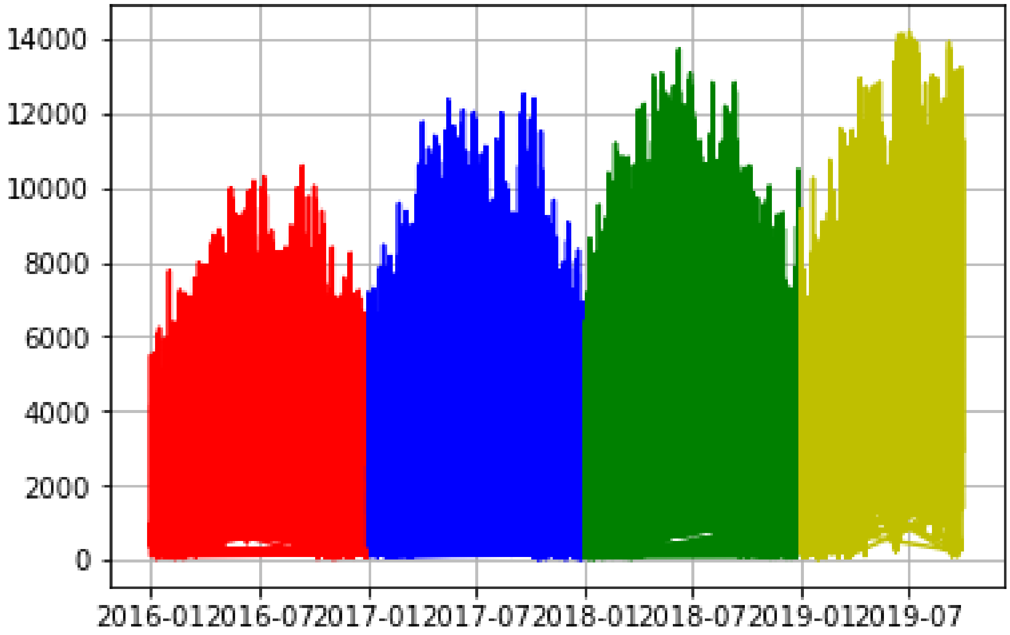


```
In [37]: df.dtypes

Out[37]: data_item              object
         value                 float64
         uom                    object
         report                 object
         mkt_type               object
         interval_end_gmt       object
         dtype: object

In [43]: df1 = sdf("select ts, value from df where data_item = 'RTM_SCHEDULE';")
         df1['ts'] = pd.to_datetime(df1['ts'])
         df1 = df1.set_index('ts')
         df1.head()

Out[43]:                       value
         ts
         2016-11-15 07:00:00  311.42
         2016-11-15 08:00:00  383.00
         2016-11-15 09:00:00  574.58
         2016-11-15 10:00:00  761.17
         2016-11-15 11:00:00  871.50

In [44]: df1.plot(grid=True)
```

```
Out[44]: <matplotlib.axes._subplots.AxesSubplot at 0x7ffa2112b320>
```



```
In [55]: import datetime
         d1 = datetime.datetime(2016,1,1)
         d2 = datetime.datetime(2017,1,1)
         d3 = datetime.datetime(2018,1,1)
         d4 = datetime.datetime(2019,1,1)
         d5 = datetime.datetime(2020,1,1)
         y1 = df1[(d1 <= df1.index) & (df1.index < d2)]
         y2 = df1[(d2 <= df1.index) & (df1.index < d3)]
         y3 = df1[(d3 <= df1.index) & (df1.index < d4)]
         y4 = df1[(d4 <= df1.index) & (df1.index < d5)]

         fig, ax = plt.subplots()
         ax.grid(True)
         plt.plot(y1, c='r')
         plt.plot(y2, c='b')
         plt.plot(y3, c='g')
         plt.plot(y4, c='y')

Out[55]: [<matplotlib.lines.Line2D at 0x7ffa21a938d0>]
```


14000
12000
10000
8000
6000
4000
2000
0
2016-01
2016-07
2017-01
2017-07
2018-01
2018-07
2019-01
2019-07