pandasVcuDF

January 22, 2025

[1]:
```
import cudf as cf
import pandas as df
```

[2]:
```
!nemreader output-csv "examples/nem12/nem12#S01#INTEGM#NEMMCO.zip"
```

```
/home/mctouch/anaconda3/envs/nem-reader/lib/python3.12/site-
packages/nemreader/outputs.py:74: FutureWarning: Series.__getitem__ treating
keys as positions is deprecated. In a future version, integer keys will always
be treated as labels (consistent with DataFrame behavior). To access a value by
position, use `ser.iloc[pos]`
  last_date = nmi_df.iloc[-1][1].strftime("%Y%m%d")
Created NEM1201006_20040305_transposed.csv
```

[3]:
```
%%time
from nemreader import NEMFile
m = NEMFile('examples/unzipped/Example_NEM12_actual_interval.csv')
nemdata = m.nem_data()
print(nemdata)
```

```
<nemreader.nem_objects.NEMData object at 0x7fe976b3d640>
CPU times: user 60.3 ms, sys: 637  s, total: 60.9 ms
Wall time: 73.3 ms
```

[4]:
```
print(nemdata.header)
# HeaderRecord(version_header='NEM12', creation_date=datetime.datetime(2004, 4,␣
  ↪20, 13, 0), from_participant='MDA1', to_participant='Ret1')
```

```
HeaderRecord(version_header='NEM12', creation_date=datetime.datetime(2004, 5, 1,
11, 35), from_participant='MDA1', to_participant='Ret1',
file_name='examples/unzipped/Example_NEM12_actual_interval.csv', assumed=False)
```

[5]:
```
print(nemdata.transactions)
# {'VABD000163': {'E1': [], 'Q1': []}}
```

```
{'VABD000163': {'E1': [], 'Q1': []}}
```

[6]:
```
%%time
from nemreader import NEMFile
m = NEMFile('examples/unzipped/Example_NEM12_actual_interval.csv')
df = m.get_data_frame()
```

```
print(df)
```

```
           nmi suffix      serno              t_start                t_end  \
0   VABD000163     E1  METSER123 2004-02-01 00:00:00 2004-02-01 00:30:00
1   VABD000163     E1  METSER123 2004-02-01 00:30:00 2004-02-01 01:00:00
2   VABD000163     E1  METSER123 2004-02-01 01:00:00 2004-02-01 01:30:00
3   VABD000163     E1  METSER123 2004-02-01 01:30:00 2004-02-01 02:00:00
4   VABD000163     E1  METSER123 2004-02-01 02:00:00 2004-02-01 02:30:00
..         …    …        …                  …                  …
43  VABD000163     Q1  METSER123 2004-02-01 21:30:00 2004-02-01 22:00:00
44  VABD000163     Q1  METSER123 2004-02-01 22:00:00 2004-02-01 22:30:00
45  VABD000163     Q1  METSER123 2004-02-01 22:30:00 2004-02-01 23:00:00
46  VABD000163     Q1  METSER123 2004-02-01 23:00:00 2004-02-01 23:30:00
47  VABD000163     Q1  METSER123 2004-02-01 23:30:00 2004-02-02 00:00:00

    value quality evt_code evt_desc
0   1.111       A
1   1.111       A
2   1.111       A
3   1.111       A
4   1.111       A
..    …     …       …       …
43  2.222       A
44  2.222       A
45  2.222       A
46  2.222       A
47  2.222       A

[96 rows x 9 columns]
CPU times: user 17.4 ms, sys: 0 ns, total: 17.4 ms
Wall time: 31.2 ms
```

[7]:
```
%%time
from nemreader import NEMFile
m = NEMFile('examples/unzipped/Example_NEM12_actual_interval.csv')
cf = m.get_data_frame()
print(cf)
```

```
           nmi suffix      serno              t_start                t_end  \
0   VABD000163     E1  METSER123 2004-02-01 00:00:00 2004-02-01 00:30:00
1   VABD000163     E1  METSER123 2004-02-01 00:30:00 2004-02-01 01:00:00
2   VABD000163     E1  METSER123 2004-02-01 01:00:00 2004-02-01 01:30:00
3   VABD000163     E1  METSER123 2004-02-01 01:30:00 2004-02-01 02:00:00
4   VABD000163     E1  METSER123 2004-02-01 02:00:00 2004-02-01 02:30:00
..         …    …        …                  …                  …
43  VABD000163     Q1  METSER123 2004-02-01 21:30:00 2004-02-01 22:00:00
44  VABD000163     Q1  METSER123 2004-02-01 22:00:00 2004-02-01 22:30:00
45  VABD000163     Q1  METSER123 2004-02-01 22:30:00 2004-02-01 23:00:00
```

```
46  VABD000163     Q1  METSER123 2004-02-01 23:00:00 2004-02-01 23:30:00
47  VABD000163     Q1  METSER123 2004-02-01 23:30:00 2004-02-02 00:00:00

    value quality evt_code evt_desc
0   1.111       A
1   1.111       A
2   1.111       A
3   1.111       A
4   1.111       A
..    …     …       …       …
43  2.222       A
44  2.222       A
45  2.222       A
46  2.222       A
47  2.222       A

[96 rows x 9 columns]
CPU times: user 11.9 ms, sys: 0 ns, total: 11.9 ms
Wall time: 10.6 ms
```

[8]:
```
df = m.get_pivot_data_frame()
print(df)
```

```
           nmi              t_start                t_end     E1     Q1 quality  \
0   VABD000163 2004-02-01 00:00:00 2004-02-01 00:30:00  1.111  2.222       A
1   VABD000163 2004-02-01 00:30:00 2004-02-01 01:00:00  1.111  2.222       A
2   VABD000163 2004-02-01 01:00:00 2004-02-01 01:30:00  1.111  2.222       A
3   VABD000163 2004-02-01 01:30:00 2004-02-01 02:00:00  1.111  2.222       A
4   VABD000163 2004-02-01 02:00:00 2004-02-01 02:30:00  1.111  2.222       A
5   VABD000163 2004-02-01 02:30:00 2004-02-01 03:00:00  1.111  2.222       A
6   VABD000163 2004-02-01 03:00:00 2004-02-01 03:30:00  1.111  2.222       A
7   VABD000163 2004-02-01 03:30:00 2004-02-01 04:00:00  1.111  2.222       A
8   VABD000163 2004-02-01 04:00:00 2004-02-01 04:30:00  1.111  2.222       A
9   VABD000163 2004-02-01 04:30:00 2004-02-01 05:00:00  1.111  2.222       A
10  VABD000163 2004-02-01 05:00:00 2004-02-01 05:30:00  1.111  2.222       A
11  VABD000163 2004-02-01 05:30:00 2004-02-01 06:00:00  1.111  2.222       A
12  VABD000163 2004-02-01 06:00:00 2004-02-01 06:30:00  1.111  2.222       A
13  VABD000163 2004-02-01 06:30:00 2004-02-01 07:00:00  1.111  2.222       A
14  VABD000163 2004-02-01 07:00:00 2004-02-01 07:30:00  1.111  2.222       A
15  VABD000163 2004-02-01 07:30:00 2004-02-01 08:00:00  1.111  2.222       A
16  VABD000163 2004-02-01 08:00:00 2004-02-01 08:30:00  1.111  2.222       A
17  VABD000163 2004-02-01 08:30:00 2004-02-01 09:00:00  1.111  2.222       A
18  VABD000163 2004-02-01 09:00:00 2004-02-01 09:30:00  1.111  2.222       A
19  VABD000163 2004-02-01 09:30:00 2004-02-01 10:00:00  1.111  2.222       A
20  VABD000163 2004-02-01 10:00:00 2004-02-01 10:30:00  1.111  2.222       A
21  VABD000163 2004-02-01 10:30:00 2004-02-01 11:00:00  1.111  2.222       A
22  VABD000163 2004-02-01 11:00:00 2004-02-01 11:30:00  1.111  2.222       A
23  VABD000163 2004-02-01 11:30:00 2004-02-01 12:00:00  1.111  2.222       A
```

24	VABD000163	2004-02-01	12:00:00	2004-02-01	12:30:00	1.111	2.222	A
25	VABD000163	2004-02-01	12:30:00	2004-02-01	13:00:00	1.111	2.222	A
26	VABD000163	2004-02-01	13:00:00	2004-02-01	13:30:00	1.111	2.222	A
27	VABD000163	2004-02-01	13:30:00	2004-02-01	14:00:00	1.111	2.222	A
28	VABD000163	2004-02-01	14:00:00	2004-02-01	14:30:00	1.111	2.222	A
29	VABD000163	2004-02-01	14:30:00	2004-02-01	15:00:00	1.111	2.222	A
30	VABD000163	2004-02-01	15:00:00	2004-02-01	15:30:00	1.111	2.222	A
31	VABD000163	2004-02-01	15:30:00	2004-02-01	16:00:00	1.111	2.222	A
32	VABD000163	2004-02-01	16:00:00	2004-02-01	16:30:00	1.111	2.222	A
33	VABD000163	2004-02-01	16:30:00	2004-02-01	17:00:00	1.111	2.222	A
34	VABD000163	2004-02-01	17:00:00	2004-02-01	17:30:00	1.111	2.222	A
35	VABD000163	2004-02-01	17:30:00	2004-02-01	18:00:00	1.111	2.222	A
36	VABD000163	2004-02-01	18:00:00	2004-02-01	18:30:00	1.111	2.222	A
37	VABD000163	2004-02-01	18:30:00	2004-02-01	19:00:00	1.111	2.222	A
38	VABD000163	2004-02-01	19:00:00	2004-02-01	19:30:00	1.111	2.222	A
39	VABD000163	2004-02-01	19:30:00	2004-02-01	20:00:00	1.111	2.222	A
40	VABD000163	2004-02-01	20:00:00	2004-02-01	20:30:00	1.111	2.222	A
41	VABD000163	2004-02-01	20:30:00	2004-02-01	21:00:00	1.111	2.222	A
42	VABD000163	2004-02-01	21:00:00	2004-02-01	21:30:00	1.111	2.222	A
43	VABD000163	2004-02-01	21:30:00	2004-02-01	22:00:00	1.111	2.222	A
44	VABD000163	2004-02-01	22:00:00	2004-02-01	22:30:00	1.111	2.222	A
45	VABD000163	2004-02-01	22:30:00	2004-02-01	23:00:00	1.111	2.222	A
46	VABD000163	2004-02-01	23:00:00	2004-02-01	23:30:00	1.111	2.222	A
47	VABD000163	2004-02-01	23:30:00	2004-02-02	00:00:00	1.111	2.222	A

	evt_code	evt_desc
0		
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
20		
21		

[9]:
```
df = m.get_pivot_data_frame()
print(df)
```

```
           nmi              t_start                t_end     E1     Q1 quality  \
0   VABD000163 2004-02-01 00:00:00 2004-02-01 00:30:00  1.111  2.222       A
1   VABD000163 2004-02-01 00:30:00 2004-02-01 01:00:00  1.111  2.222       A
2   VABD000163 2004-02-01 01:00:00 2004-02-01 01:30:00  1.111  2.222       A
3   VABD000163 2004-02-01 01:30:00 2004-02-01 02:00:00  1.111  2.222       A
4   VABD000163 2004-02-01 02:00:00 2004-02-01 02:30:00  1.111  2.222       A
5   VABD000163 2004-02-01 02:30:00 2004-02-01 03:00:00  1.111  2.222       A
6   VABD000163 2004-02-01 03:00:00 2004-02-01 03:30:00  1.111  2.222       A
7   VABD000163 2004-02-01 03:30:00 2004-02-01 04:00:00  1.111  2.222       A
8   VABD000163 2004-02-01 04:00:00 2004-02-01 04:30:00  1.111  2.222       A
9   VABD000163 2004-02-01 04:30:00 2004-02-01 05:00:00  1.111  2.222       A
10  VABD000163 2004-02-01 05:00:00 2004-02-01 05:30:00  1.111  2.222       A
11  VABD000163 2004-02-01 05:30:00 2004-02-01 06:00:00  1.111  2.222       A
12  VABD000163 2004-02-01 06:00:00 2004-02-01 06:30:00  1.111  2.222       A
13  VABD000163 2004-02-01 06:30:00 2004-02-01 07:00:00  1.111  2.222       A
14  VABD000163 2004-02-01 07:00:00 2004-02-01 07:30:00  1.111  2.222       A
15  VABD000163 2004-02-01 07:30:00 2004-02-01 08:00:00  1.111  2.222       A
16  VABD000163 2004-02-01 08:00:00 2004-02-01 08:30:00  1.111  2.222       A
```

17	VABD000163	2004-02-01	08:30:00	2004-02-01	09:00:00	1.111	2.222	A
18	VABD000163	2004-02-01	09:00:00	2004-02-01	09:30:00	1.111	2.222	A
19	VABD000163	2004-02-01	09:30:00	2004-02-01	10:00:00	1.111	2.222	A
20	VABD000163	2004-02-01	10:00:00	2004-02-01	10:30:00	1.111	2.222	A
21	VABD000163	2004-02-01	10:30:00	2004-02-01	11:00:00	1.111	2.222	A
22	VABD000163	2004-02-01	11:00:00	2004-02-01	11:30:00	1.111	2.222	A
23	VABD000163	2004-02-01	11:30:00	2004-02-01	12:00:00	1.111	2.222	A
24	VABD000163	2004-02-01	12:00:00	2004-02-01	12:30:00	1.111	2.222	A
25	VABD000163	2004-02-01	12:30:00	2004-02-01	13:00:00	1.111	2.222	A
26	VABD000163	2004-02-01	13:00:00	2004-02-01	13:30:00	1.111	2.222	A
27	VABD000163	2004-02-01	13:30:00	2004-02-01	14:00:00	1.111	2.222	A
28	VABD000163	2004-02-01	14:00:00	2004-02-01	14:30:00	1.111	2.222	A
29	VABD000163	2004-02-01	14:30:00	2004-02-01	15:00:00	1.111	2.222	A
30	VABD000163	2004-02-01	15:00:00	2004-02-01	15:30:00	1.111	2.222	A
31	VABD000163	2004-02-01	15:30:00	2004-02-01	16:00:00	1.111	2.222	A
32	VABD000163	2004-02-01	16:00:00	2004-02-01	16:30:00	1.111	2.222	A
33	VABD000163	2004-02-01	16:30:00	2004-02-01	17:00:00	1.111	2.222	A
34	VABD000163	2004-02-01	17:00:00	2004-02-01	17:30:00	1.111	2.222	A
35	VABD000163	2004-02-01	17:30:00	2004-02-01	18:00:00	1.111	2.222	A
36	VABD000163	2004-02-01	18:00:00	2004-02-01	18:30:00	1.111	2.222	A
37	VABD000163	2004-02-01	18:30:00	2004-02-01	19:00:00	1.111	2.222	A
38	VABD000163	2004-02-01	19:00:00	2004-02-01	19:30:00	1.111	2.222	A
39	VABD000163	2004-02-01	19:30:00	2004-02-01	20:00:00	1.111	2.222	A
40	VABD000163	2004-02-01	20:00:00	2004-02-01	20:30:00	1.111	2.222	A
41	VABD000163	2004-02-01	20:30:00	2004-02-01	21:00:00	1.111	2.222	A
42	VABD000163	2004-02-01	21:00:00	2004-02-01	21:30:00	1.111	2.222	A
43	VABD000163	2004-02-01	21:30:00	2004-02-01	22:00:00	1.111	2.222	A
44	VABD000163	2004-02-01	22:00:00	2004-02-01	22:30:00	1.111	2.222	A
45	VABD000163	2004-02-01	22:30:00	2004-02-01	23:00:00	1.111	2.222	A
46	VABD000163	2004-02-01	23:00:00	2004-02-01	23:30:00	1.111	2.222	A
47	VABD000163	2004-02-01	23:30:00	2004-02-02	00:00:00	1.111	2.222	A

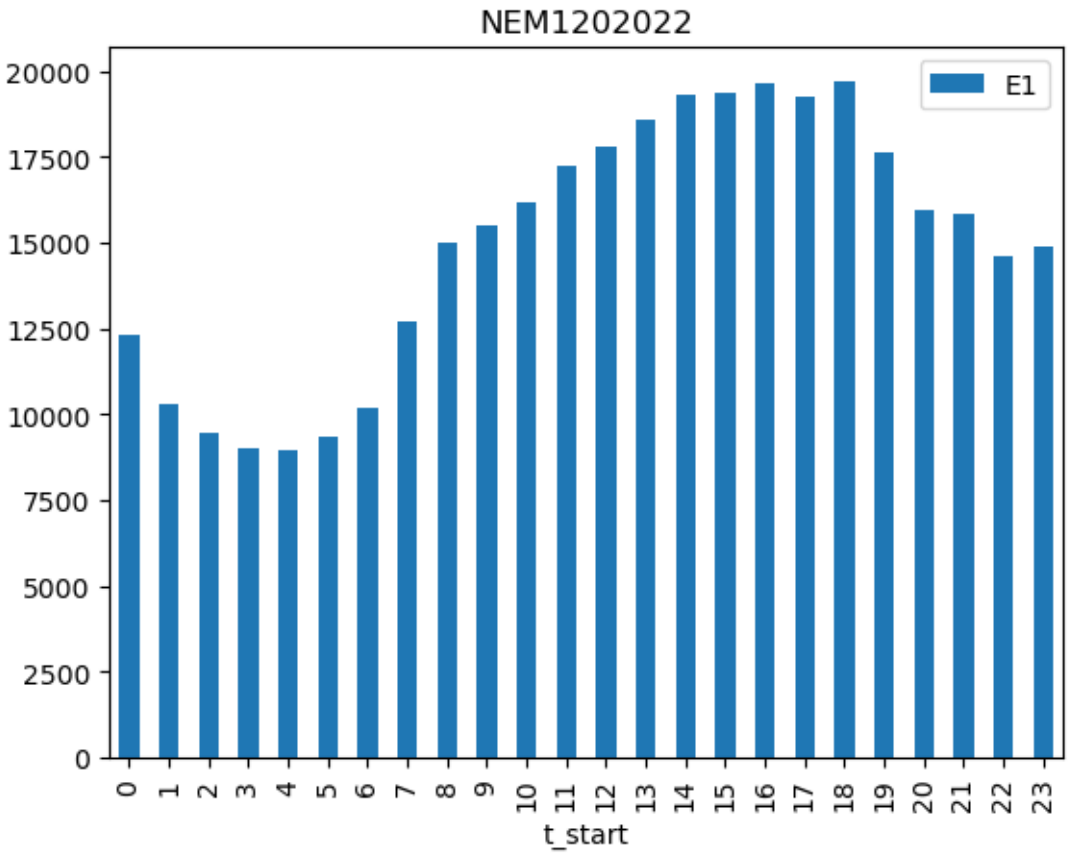
	evt_code	evt_desc
0		
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		

[10]:
```
%%time
import matplotlib.pyplot as plt
from nemreader import output_as_data_frames

# Setup Pandas DataFrame
dfs = output_as_data_frames("examples/nem12/
  ↪NEM12#000000000000002#CNRGYMDP#NEMMCO.zip")
nmi, df = dfs[0] # Return data for first NMI in file
df.set_index("t_start", inplace=True)

# Chart time of day profile
hourly = df.groupby([(df.index.hour)]).sum(numeric_only=True)
plot = hourly.plot(title=nmi, kind="bar", y=["E1"])
```

```
plt.show()
```

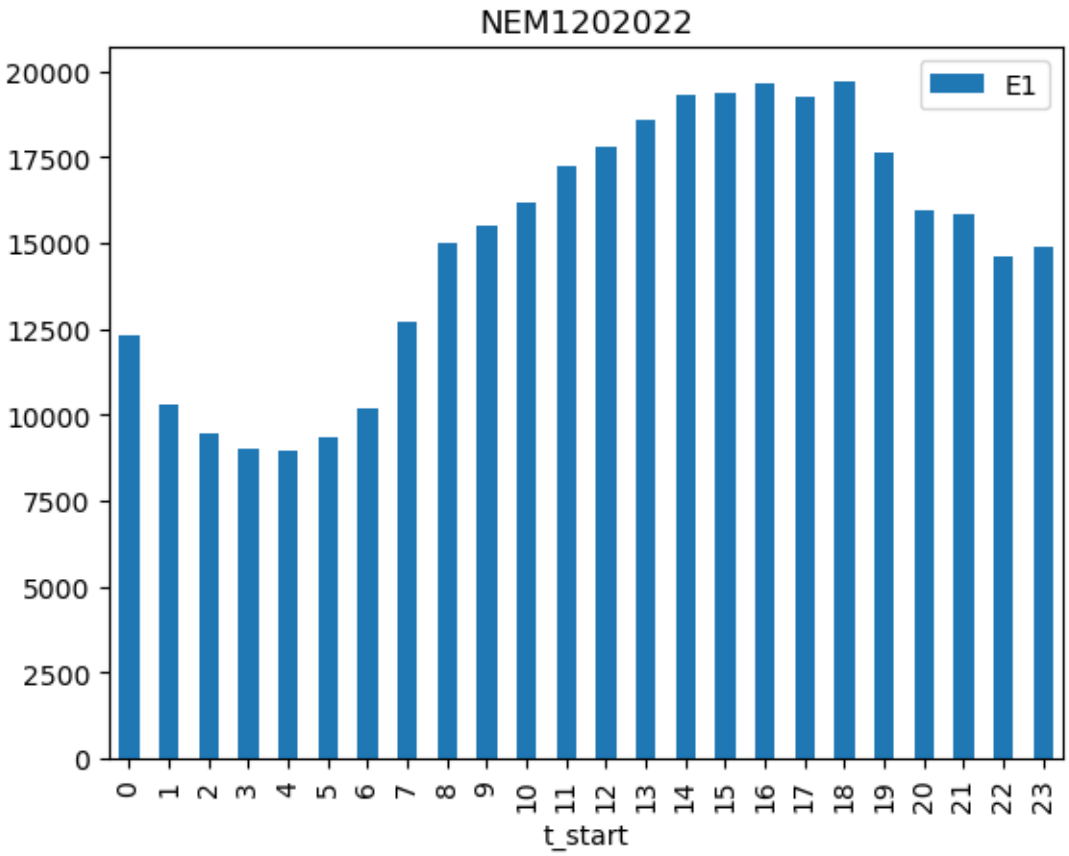



```
CPU times: user 693 ms, sys: 48 ms, total: 741 ms
Wall time: 1.62 s
```

[11]:
```
%%time
import matplotlib.pyplot as plt
from nemreader import output_as_cudf_data_frames

# Setup Pandas DataFrame
dfs = output_as_data_frames("examples/nem12/
  ↪NEM12#000000000000002#CNRGYMDP#NEMMCO.zip")
nmi, df = dfs[0] # Return data for first NMI in file
df.set_index("t_start", inplace=True)

# Chart time of day profile
hourly = df.groupby([(df.index.hour)]).sum(numeric_only=True)
plot = hourly.plot(title=nmi, kind="bar", y=["E1"])
plt.show()
```




```
CPU times: user 226 ms, sys: 372  s, total: 227 ms
Wall time: 217 ms
```

[ ]: