In [1]:

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
%matplotlib inline
#plt.style.use('seaborn')
```

In [2]:

```
sns.set_style("ticks")
```

Preparing the dataframes

In [4]:

```
df_all = pd.read_csv('dataport-export_gas_oct2015-mar2016.csv')
len(df_all)
```

Out[4]:

```
1584823
```

In [5]:

```
df_all = df_all.set_index(pd.to_datetime(df_all['localminute']))
```

In [6]:

```
display(df_all.head(), df_all.tail())
```

	localminute	dataid	meter_value
localminute			
2015-10-01 00:00:10-05:00	2015-10-01 00:00:10-05	739	88858
2015-10-01 00:00:13-05:00	2015-10-01 00:00:13-05	8890	197164
2015-10-01 00:00:20-05:00	2015-10-01 00:00:20-05	6910	179118
2015-10-01 00:00:22-05:00	2015-10-01 00:00:22-05	3635	151318
2015-10-01 00:00:22-05:00	2015-10-01 00:00:22-05	1507	390354

	localminute	dataid	meter_value
localminute			
2016-03-31 23:59:14.336743-05:00	2016-03-31 23:59:14.336743-05	2129	201726
2016-03-31 23:59:17.427165-05:00	2016-03-31 23:59:17.427165-05	2945	161232
2016-03-31 23:59:35.370782-05:00	2016-03-31 23:59:35.370782-05	9729	138146
2016-03-31 23:59:47.816286-05:00	2016-03-31 23:59:47.816286-05	5129	166488
2016-03-31 23:59:58.923080-05:00	2016-03-31 23:59:58.92308-05	484	114174

In [7]:

```
df_all = df_all.drop(columns='localminute')
display(df_all.head())
```

	dataid	meter_value
localminute		
2015-10-01 00:00:10-05:00	739	88858
2015-10-01 00:00:13-05:00	8890	197164
2015-10-01 00:00:20-05:00	6910	179118
2015-10-01 00:00:22-05:00	3635	151318
2015-10-01 00:00:22-05:00	1507	390354

In [8]:

```
groups = df_all.groupby('dataid')
keys = groups.groups.keys()  # keys: an iterable of dataids or meter ids

# check if each group (grouped by meter id) is sorted in ascending order by datetime.
for key in keys:
    df_i = groups.get_group(key)
    if df_i.index.is_monotonic_increasing is False:
        print(key)
```

Check meterids

In [9]:

```
keys_list = list(keys)
print(keys_list)
```

```
[35, 44, 77, 94, 114, 187, 222, 252, 370, 483, 484, 661, 739, 744, 871, 1042,
1086, 1103, 1185, 1283, 1403, 1415, 1507, 1556, 1589, 1619, 1697, 1714, 1718,
1790, 1791, 1792, 1800, 1801, 2018, 2034, 2072, 2094, 2129, 2233, 2335, 2378,
2449, 2461, 2470, 2575, 2638, 2645, 2755, 2814, 2818, 2945, 2946, 2965, 2980,
3036, 3039, 3134, 3310, 3367, 3527, 3544, 3577, 3635, 3723, 3778, 3849, 3893,
3918, 4029, 4031, 4193, 4228, 4296, 4352, 4356, 4373, 4421, 4447, 4514, 4671,
4732, 4767, 4874, 4998, 5129, 5131, 5193, 5275, 5317, 5395, 5403, 5439, 5484,
5545, 5636, 5658, 5785, 5810, 5814, 5892, 5972, 6101, 6412, 6505, 6578, 6673,
6685, 6830, 6836, 6863, 6910, 7016, 7017, 7030, 7117, 7287, 7429, 7460, 7566,
7674, 7682, 7739, 7741, 7794, 7900, 7919, 7965, 7989, 8059, 8084, 8086, 8155,
8156, 8244, 8386, 8467, 8703, 8829, 8890, 8967, 9052, 9121, 9134, 9160, 9278,
9295, 9474, 9600, 9620, 9631, 9639, 9729, 9766, 9849, 9956, 9982]
```

Check data count per meter

In [12]:

```
count_list = []
for key in keys_list:
    df_i = groups.get_group(key)
    count_list.append(len(df_i.index))

print(count_list)
```

```
[11872, 1549, 10683, 36335, 2597, 914, 2731, 16774, 3641, 27628, 44034, 3622,
31430, 6058, 35070, 3830, 30029, 696, 18456, 12228, 202, 930, 32603, 3690, 26
352, 2983, 4690, 32933, 24470, 13344, 11060, 1646, 5590, 15892, 7341, 75991,
13519, 17311, 13787, 2271, 8910, 2814, 5449, 12806, 1453, 2080, 5698, 74, 68,
37, 732, 9895, 45, 5017, 3225, 336, 5400, 4017, 10200, 12068, 10853, 2221, 36
74, 9186, 8141, 13609, 1563, 26844, 6325, 10356, 12534, 1019, 799, 1176, 330
4, 1924, 1692, 3269, 9158, 19074, 21, 5303, 7583, 2, 13974, 4486, 15187, 1946
4, 2289, 1039, 1545, 25559, 5972, 2056, 33, 3411, 493, 12103, 42234, 42424, 1
4139, 5243, 3, 15783, 1862, 928, 10694, 78, 2389, 4520, 672, 69349, 2929, 252
79, 17915, 20493, 41005, 13212, 3646, 32, 29329, 3391, 4433, 5644, 8529, 228
0, 2893, 641, 39723, 529, 4686, 4391, 5423, 25296, 919, 603, 9020, 258, 1488
3, 16574, 6695, 5933, 4510, 14064, 72, 6145, 26534, 3473, 330, 23, 4411, 1379
6, 12361, 2282, 2741, 1292, 1540]
```

Check culmulative reading decreases (malfunctions)

In [36]:

```
error_count_list = []
for key in keys_list:
    df_i = groups.get_group(key)
    prev_meter_value = 0
    error_count = 0
    for i in range(len(df_i.index)):
        curr_meter_value = df_i.iloc[i][1]
        if i is 0:
            prev_meter_value = curr_meter_value
        else:
            if curr_meter_value < prev_meter_value:
                error_count += 1
            prev_meter_value = curr_meter_value
    error_count_list.append(error_count)

print(error_count_list)
```

```
[1, 0, 1, 6, 0, 0, 0, 0, 0, 1, 9, 0, 0, 0, 0, 1, 1, 0, 135, 0, 0, 0, 2, 12,
0, 0, 0, 0, 4, 1, 0, 0, 0, 1, 0, 0, 0, 0, 3, 0, 5, 0, 93, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 18, 0, 0, 1, 18, 0, 0, 0, 0, 0, 2, 0, 0, 16, 1, 0, 0,
0, 0, 0, 0, 0, 141, 0, 0, 0, 0, 1, 76, 1, 4, 0, 0, 0, 156, 0, 0, 0, 0, 0, 0,
10, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 51, 0, 0, 0, 1, 90, 123, 0, 0, 0, 0, 0, 0,
1, 0, 1, 0, 0, 0, 2, 0, 0, 0, 0, 151, 0, 0, 0, 0, 0, 44, 0, 0, 0, 115, 0, 0,
0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 2]
```

In [41]:

```
percent_error_list = []
for i in range(157):
    percent_error = round(error_count_list[i] / count_list[i] * 100, 6)
    percent_error_list.append(percent_error)

print(percent_error_list)
```

```
[0.008423, 0.0, 0.009361, 0.016513, 0.0, 0.0, 0.0, 0.0, 0.0, 0.00362, 0.02043
9, 0.0, 0.0, 0.0, 0.0, 0.02611, 0.00333, 0.0, 0.731469, 0.0, 0.0, 0.0, 0.0061
34, 0.325203, 0.0, 0.0, 0.0, 0.0, 0.016347, 0.007494, 0.0, 0.0, 0.0, 0.00629
2, 0.0, 0.0, 0.0, 0.0, 0.02176, 0.0, 0.056117, 0.0, 1.706735, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.448096, 0.0, 0.0, 0.
009214, 0.810446, 0.0, 0.0, 0.0, 0.0, 0.0, 0.00745, 0.0, 0.0, 0.127653, 0.098
135, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.739226, 0.0, 0.0, 0.0, 0.0, 0.00715
6, 1.69416, 0.006585, 0.020551, 0.0, 0.0, 0.0, 0.610353, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.023678, 0.002357, 0.007073, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 1.128319, 0.0, 0.0, 0.0, 0.003956, 0.502372, 0.600205, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.022558, 0.0, 0.011725, 0.0, 0.0, 0.0, 0.005035, 0.0, 0.0, 0.0, 0.
0, 0.596932, 0.0, 0.0, 0.0, 0.0, 0.0, 0.265476, 0.0, 0.0, 0.0, 0.817691, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.014497, 0.0, 0.0, 0.0, 0.0, 0.12987]
```

In [46]:

```
df_error_metric = pd.DataFrame(keys_list, columns=['meterid'])
df_error_metric = df_error_metric.assign(count=count_list)
df_error_metric = df_error_metric.assign(errors=error_count_list)
df_error_metric = df_error_metric.assign(percentage=percent_error_list)
df_error_metric
```

Out[46]:

	meterid	count	errors	percentage
0	35	11872	1	0.008423
1	44	1549	0	0.000000
2	77	10683	1	0.009361
3	94	36335	6	0.016513
4	114	2597	0	0.000000
5	187	914	0	0.000000
6	222	2731	0	0.000000
7	252	16774	0	0.000000
8	370	3641	0	0.000000
9	483	27628	1	0.003620
10	484	44034	9	0.020439
11	661	3622	0	0.000000
12	739	31430	0	0.000000
13	744	6058	0	0.000000
14	871	35070	0	0.000000
15	1042	3830	1	0.026110
16	1086	30029	1	0.003330
17	1103	696	0	0.000000
18	1185	18456	135	0.731469
19	1283	12228	0	0.000000
20	1403	202	0	0.000000
21	1415	930	0	0.000000
22	1507	32603	2	0.006134
23	1556	3690	12	0.325203
24	1589	26352	0	0.000000
25	1619	2983	0	0.000000
26	1697	4690	0	0.000000
27	1714	32933	0	0.000000
28	1718	24470	4	0.016347
29	1790	13344	1	0.007494
...	...	...	...	...
127	7965	641	0	0.000000
128	7989	39723	2	0.005035
129	8059	529	0	0.000000

	meterid	count	errors	percentage
130	8084	4686	0	0.000000
131	8086	4391	0	0.000000
132	8155	5423	0	0.000000
133	8156	25296	151	0.596932
134	8244	919	0	0.000000
135	8386	603	0	0.000000
136	8467	9020	0	0.000000
137	8703	258	0	0.000000
138	8829	14883	0	0.000000
139	8890	16574	44	0.265476
140	8967	6695	0	0.000000
141	9052	5933	0	0.000000
142	9121	4510	0	0.000000
143	9134	14064	115	0.817691
144	9160	72	0	0.000000
145	9278	6145	0	0.000000
146	9295	26534	0	0.000000
147	9474	3473	0	0.000000
148	9600	330	0	0.000000
149	9620	23	0	0.000000
150	9631	4411	0	0.000000
151	9639	13796	2	0.014497
152	9729	12361	0	0.000000
153	9766	2282	0	0.000000
154	9849	2741	0	0.000000
155	9956	1292	0	0.000000
156	9982	1540	2	0.129870

157 rows × 4 columns

In [49]:

```
df_error_metric.sort_values(by=['count'])
```

Out[49]:

	meterid	count	errors	percentage
83	4874	2	0	0.000000
102	6101	3	0	0.000000
80	4671	21	0	0.000000
149	9620	23	0	0.000000
119	7566	32	0	0.000000
94	5545	33	0	0.000000
49	2814	37	0	0.000000
52	2946	45	0	0.000000
48	2755	68	0	0.000000
144	9160	72	0	0.000000
47	2645	74	0	0.000000
107	6685	78	0	0.000000
20	1403	202	0	0.000000
137	8703	258	0	0.000000
148	9600	330	0	0.000000
55	3036	336	0	0.000000
96	5658	493	0	0.000000
129	8059	529	0	0.000000
135	8386	603	0	0.000000
127	7965	641	0	0.000000
110	6863	672	0	0.000000
17	1103	696	0	0.000000
50	2818	732	0	0.000000
72	4228	799	0	0.000000
5	187	914	0	0.000000
134	8244	919	0	0.000000
105	6578	928	0	0.000000
21	1415	930	0	0.000000
71	4193	1019	1	0.098135
89	5317	1039	0	0.000000
...	...	...	...	...
139	8890	16574	44	0.265476
7	252	16774	0	0.000000
37	2094	17311	0	0.000000

	meterid	count	errors	percentage
114	7030	17915	90	0.502372
18	1185	18456	135	0.731469
79	4514	19074	141	0.739226
87	5193	19464	4	0.020551
115	7117	20493	123	0.600205
28	1718	24470	4	0.016347
113	7017	25279	1	0.003956
133	8156	25296	151	0.596932
91	5403	25559	156	0.610353
24	1589	26352	0	0.000000
146	9295	26534	0	0.000000
67	3893	26844	2	0.007450
9	483	27628	1	0.003620
120	7674	29329	0	0.000000
16	1086	30029	1	0.003330
12	739	31430	0	0.000000
22	1507	32603	2	0.006134
27	1714	32933	0	0.000000
14	871	35070	0	0.000000
3	94	36335	6	0.016513
128	7989	39723	2	0.005035
116	7287	41005	0	0.000000
98	5810	42234	10	0.023678
99	5814	42424	1	0.002357
10	484	44034	9	0.020439
111	6910	69349	0	0.000000
35	2034	75991	0	0.000000

157 rows × 4 columns

In [50]:

```
df_error_metric.sort_values(by=['percentage'], ascending=False)
```

Out[50]:

	meterid	count	errors	percentage
42	2449	5449	93	1.706735
85	5129	4486	76	1.694160
109	6836	4520	51	1.128319
143	9134	14064	115	0.817691
61	3544	2221	18	0.810446
79	4514	19074	141	0.739226
18	1185	18456	135	0.731469
91	5403	25559	156	0.610353
115	7117	20493	123	0.600205
133	8156	25296	151	0.596932
114	7030	17915	90	0.502372
57	3134	4017	18	0.448096
23	1556	3690	12	0.325203
139	8890	16574	44	0.265476
156	9982	1540	2	0.129870
70	4031	12534	16	0.127653
71	4193	1019	1	0.098135
40	2335	8910	5	0.056117
15	1042	3830	1	0.026110
98	5810	42234	10	0.023678
122	7739	4433	1	0.022558
38	2129	13787	3	0.021760
87	5193	19464	4	0.020551
10	484	44034	9	0.020439
3	94	36335	6	0.016513
28	1718	24470	4	0.016347
151	9639	13796	2	0.014497
124	7794	8529	1	0.011725
2	77	10683	1	0.009361
60	3527	10853	1	0.009214
...	...	...	...	...
63	3635	9186	0	0.000000
96	5658	493	0	0.000000
80	4671	21	0	0.000000

	meterid	count	errors	percentage
95	5636	3411	0	0.000000
32	1800	5590	0	0.000000
93	5484	2056	0	0.000000
92	5439	5972	0	0.000000
90	5395	1545	0	0.000000
89	5317	1039	0	0.000000
88	5275	2289	0	0.000000
17	1103	696	0	0.000000
19	1283	12228	0	0.000000
83	4874	2	0	0.000000
82	4767	7583	0	0.000000
81	4732	5303	0	0.000000
20	1403	202	0	0.000000
64	3723	8141	0	0.000000
1	44	1549	0	0.000000
77	4421	3269	0	0.000000
76	4373	1692	0	0.000000
75	4356	1924	0	0.000000
74	4352	3304	0	0.000000
73	4296	1176	0	0.000000
72	4228	799	0	0.000000
21	1415	930	0	0.000000
69	4029	10356	0	0.000000
68	3918	6325	0	0.000000
66	3849	1563	0	0.000000
65	3778	13609	0	0.000000
78	4447	9158	0	0.000000

157 rows × 4 columns

In [87]:

```
df_error_metric_sorted = df_error_metric.sort_values(by=['percentage'], ascending=False)
keys_list2 = df_error_metric_sorted["meterid"].tolist()
keys_list2 = keys_list2[18:]
```

Checking culmulative reading sharp increases (potential malfunction)

In [88]:

```
average_increment_list = []

for key in keys_list2:
    df_i = groups.get_group(key)
    prev_meter_value = 0
    sum_increment = 0
    sum_count = 0
    for i in range(len(df_i.index)):
        curr_meter_value = df_i.iloc[i][1]
        if i != 0:
            if curr_meter_value > prev_meter_value:
                sum_increment += (curr_meter_value - prev_meter_value)
                sum_count += 1
        prev_meter_value = curr_meter_value
    average_increment = round(sum_increment / sum_count, 6)
    average_increment_list.append(average_increment)

print(average_increment_list)
```

```
[10.42692, 3.94857, 6.987342, 6.316891, 4.386805, 3.005447, 4.651663, 5.73517
6, 18.584844, 8.742521, 4.071329, 6.862478, 5.094871, 12.15355, 5.505476, 7.3
90728, 5.681843, 5.994563, 6.793639, 5.756627, 3.738176, 6.925804, 5.786694,
4.887492, 3.587503, 244.101695, 3.969605, 12.521146, 15.581065, 671.0, 7.6197
04, 8.667235, 4.791174, 113.666667, 20.403064, 8.870763, 4.23814, 14.485459,
3.998305, 36.7875, 21.560821, 3.328149, 6.844444, 8.909443, 17.276923, 12.910
288, 140.697674, 13.177066, 17.051656, 9.320015, 24.361656, 8.202899, 30.7327
59, 5.747209, 12.813866, 11.101031, 1698.0, 12.854962, 19.767045, 4.620885,
6.796938, 65.1875, 31.607397, 12.11974, 4.700599, 3.471452, 69.313496, 5.7715
6, 92.021622, 6.590345, 24.393204, 6.771372, 7.172503, 13.294375, 13.77413,
9.859155, 34.51634, 54.532934, 13.586796, 329.741935, 10.779982, 81.612903,
8.841008, 6.745544, 6.569719, 10.733582, 13.59612, 17.582222, 8.662338, 7.062
817, 113.952381, 4.616251, 35.869416, 218.882353, 143.827586, 11.758442, 15.7
4145, 20.8879, 17.993769, 7.651042, 5.768904, 28.029205, 34.030418, 5.840091,
4.574045, 8.610949, 3.229239, 16.232798, 8.566618, 5.823775, 26.951111, 155.8
94737, 10.470128, 11.210637, 15.312916, 8.222958, 19.85124, 3.655738, 24.7977
84, 36.617414, 5.594539, 6986.0, 13.581808, 10.065766, 15.836066, 8.663411, 2
1.403189, 10.338756, 19.900344, 19.870085, 22.982659, 24.935393, 20.062992, 3
2.483636, 14.371047, 14.70216, 22.381955, 6.792435, 6.159564]
```

In [90]:

```
delta_1percent_list = []
delta_5percent_list = []
delta_10percent_list = []
delta_20percent_list = []
delta_50percent_list = []
constant_list = []
decrease_list = []

for key in keys_list2:
    df_i = groups.get_group(key)
    prev_meter_value = 0
    delta_1percent_count = 0
    delta_5percent_count = 0
    delta_10percent_count = 0
    delta_20percent_count = 0
    delta_50percent_count = 0
    constant_count = 0
    decrease_count = 0
    average_increment = average_increment_list[keys_list2.index(key)]
    for i in range(len(df_i.index)):
        curr_meter_value = df_i.iloc[i][1]
        if i != 0:
            if curr_meter_value > prev_meter_value:
                increment = curr_meter_value - prev_meter_value
                delta_percentage = round(abs(increment - average_increment) / average_incr
ement * 100, 3)
                if delta_percentage <= 1:
                    delta_1percent_count += 1
                elif delta_percentage <= 5:
                    delta_5percent_count += 1
                elif delta_percentage <= 10:
                    delta_10percent_count += 1
                elif delta_percentage <= 20:
                    delta_20percent_count += 1
                else:
                    delta_50percent_count += 1
            elif curr_meter_value == prev_meter_value:
                constant_count += 1
            else:
                decrease_count += 1

        prev_meter_value = curr_meter_value

    delta_1percent_list.append(delta_1percent_count)
    delta_5percent_list.append(delta_5percent_count)
    delta_10percent_list.append(delta_10percent_count)
    delta_20percent_list.append(delta_20percent_count)
    delta_50percent_list.append(delta_50percent_count)
    constant_list.append(constant_count)
    decrease_list.append(decrease_count)
```

In [91]:

```
df_increase_error = pd.DataFrame(keys_list2, columns=['meterid'])
df_increase_error = df_increase_error.assign(averageIncrement=average_increment_list)
df_increase_error = df_increase_error.assign(within1percent=delta_1percent_list)
df_increase_error = df_increase_error.assign(within5percent=delta_5percent_list)
df_increase_error = df_increase_error.assign(within10percent=delta_10percent_list)
df_increase_error = df_increase_error.assign(within20percent=delta_20percent_list)
df_increase_error = df_increase_error.assign(within50percent=delta_50percent_list)
df_increase_error = df_increase_error.assign(constant=constant_list)
df_increase_error = df_increase_error.assign(decrease=decrease_list)
df_increase_error
```

Out[91]:

	meterid	averageIncrement	within1percent	within5percent	within10percent	within20percent	v
0	1042	10.426920	0	44	0	37	
1	5810	3.948570	0	881	0	0	
2	7739	6.987342	0	0	0	276	
3	2129	6.316891	0	0	691	0	
4	5193	4.386805	0	0	688	0	
5	484	3.005447	0	0	0	0	
6	94	4.651663	0	0	0	1069	
7	1718	5.735176	0	345	0	0	
8	9639	18.584844	0	97	64	228	
9	7794	8.742521	0	0	265	176	
10	77	4.071329	0	268	0	0	
11	3527	6.862478	0	0	0	541	
12	35	5.094871	0	0	0	249	
13	1790	12.153550	0	293	0	489	
14	3893	5.505476	0	0	286	0	
15	4998	7.390728	0	0	269	411	
16	5892	5.681843	0	0	262	0	
17	5131	5.994563	362	0	0	0	
18	1801	6.793639	0	0	0	303	
19	1507	5.756627	0	672	0	0	
20	7989	3.738176	0	0	760	0	
21	7017	6.925804	0	0	0	1013	
22	483	5.786694	0	573	0	0	
23	1086	4.887492	0	0	0	760	
24	5814	3.587503	0	0	0	543	
25	6685	244.101695	0	0	1	3	
26	6673	3.969605	271	0	0	0	
27	7741	12.521146	0	45	0	51	
28	5972	15.581065	0	83	0	230	
29	6101	671.000000	0	0	0	0	
...	...	...	...	...	...	...	
109	3635	5.823775	0	250	0	0	
110	5658	26.951111	0	8	0	7	
111	4671	155.894737	0	0	1	0	

	meterid	averageIncrement	within1percent	within5percent	within10percent	within20percent	v
112	5636	10.470128	0	129	0	90	
113	1800	11.210637	0	0	27	38	
114	5484	15.312916	0	43	46	45	
115	5439	8.222958	0	143	0	0	
116	5395	19.851240	22	0	24	54	
117	5317	3.655738	0	0	20	0	
118	5275	24.797784	0	30	0	43	
119	1103	36.617414	0	5	8	12	
120	1283	5.594539	0	0	413	0	
121	4874	6986.000000	1	0	0	0	
122	4767	13.581808	0	51	0	95	
123	4732	10.065766	81	0	0	68	
124	1403	15.836066	0	2	0	0	
125	3723	8.663411	0	0	221	181	
126	44	21.403189	0	6	9	10	
127	4421	10.338756	0	53	0	35	
128	4373	19.900344	32	0	26	55	
129	4356	19.870085	52	0	57	84	
130	4352	22.982659	0	22	0	13	
131	4296	24.935393	0	42	0	42	
132	4228	20.062992	14	0	8	27	
133	1415	32.483636	0	17	6	42	
134	4029	14.371047	0	107	0	183	
135	3918	14.702160	0	97	85	167	
136	3849	22.381955	0	19	13	61	
137	3778	6.792435	0	0	0	492	
138	4447	6.159564	0	138	0	0	

139 rows × 9 columns

In [92]:

```
df_increase_error.sort_values(by=['averageIncrement'], ascending=False)
```

Out[92]:

	meterid	averageIncrement	within1percent	within5percent	within10percent	within20percent	v
121	4874	6986.000000	1	0	0	0	
56	9620	1698.000000	0	0	0	0	
29	6101	671.000000	0	0	0	0	
79	5545	329.741935	0	2	1	0	
25	6685	244.101695	0	0	1	3	
93	2814	218.882353	0	0	0	2	
111	4671	155.894737	0	0	1	0	
94	2645	143.827586	0	0	0	1	
46	6863	140.697674	0	1	1	6	
90	2946	113.952381	0	1	1	5	
33	7566	113.666667	0	0	2	1	
68	8703	92.021622	1	1	1	5	
81	2755	81.612903	1	0	0	1	
66	222	69.313496	5	3	4	13	
61	9160	65.187500	0	2	1	1	
77	7965	54.532934	1	3	1	9	
39	6505	36.787500	0	6	9	18	
119	1103	36.617414	0	5	8	12	
92	2818	35.869416	8	0	9	20	
76	8059	34.516340	0	4	3	20	
102	2233	34.030418	5	0	14	27	
133	1415	32.483636	0	17	6	42	
62	187	31.607397	0	6	18	32	
52	9766	30.732759	0	6	5	11	
101	2378	28.029205	29	0	66	87	
110	5658	26.951111	0	8	0	7	
131	4296	24.935393	0	42	0	42	
118	5275	24.797784	0	30	0	43	
70	8386	24.393204	0	8	15	16	
50	9956	24.361656	0	7	6	15	
...	...	...	...	...	...	...	
22	483	5.786694	0	573	0	0	
67	8829	5.771560	0	365	0	0	
100	1714	5.768904	0	435	0	0	

	meterid	averageIncrement	within1percent	within5percent	within10percent	within20percent	v
19	1507	5.756627	0	672	0	0	
53	9729	5.747209	0	290	0	0	
7	1718	5.735176	0	345	0	0	
16	5892	5.681843	0	0	262	0	
120	1283	5.594539	0	0	413	0	
14	3893	5.505476	0	0	286	0	
12	35	5.094871	0	0	0	249	
23	1086	4.887492	0	0	0	760	
32	7674	4.791174	0	0	0	1099	
64	871	4.700599	0	0	0	726	
6	94	4.651663	0	0	0	1069	
59	9295	4.620885	0	0	0	1130	
91	2945	4.616251	0	0	0	530	
104	2094	4.574045	0	0	0	869	
4	5193	4.386805	0	0	688	0	
36	7287	4.238140	0	0	833	0	
10	77	4.071329	0	268	0	0	
38	739	3.998305	666	0	0	0	
26	6673	3.969605	271	0	0	0	
1	5810	3.948570	0	881	0	0	
20	7989	3.738176	0	0	760	0	
117	5317	3.655738	0	0	20	0	
24	5814	3.587503	0	0	0	543	
65	8967	3.471452	0	0	0	19	
41	6910	3.328149	0	0	0	0	
106	2034	3.229239	0	0	0	0	
5	484	3.005447	0	0	0	0	

139 rows × 9 columns

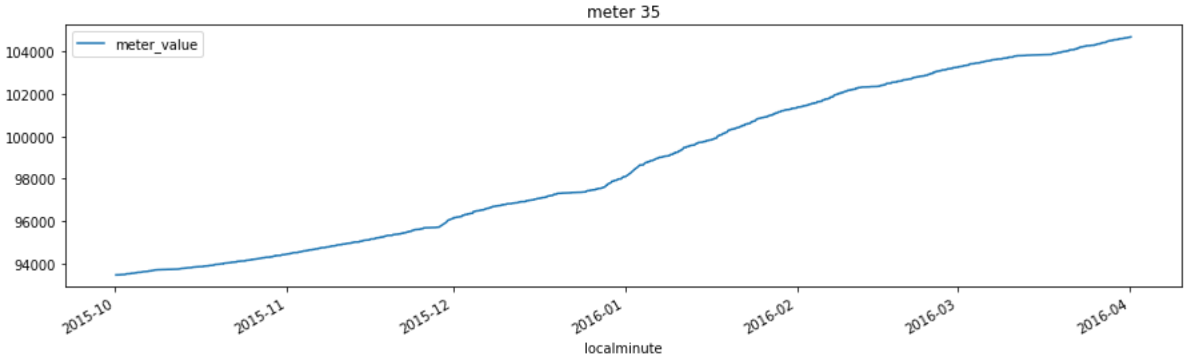
```
In [ ]:
```

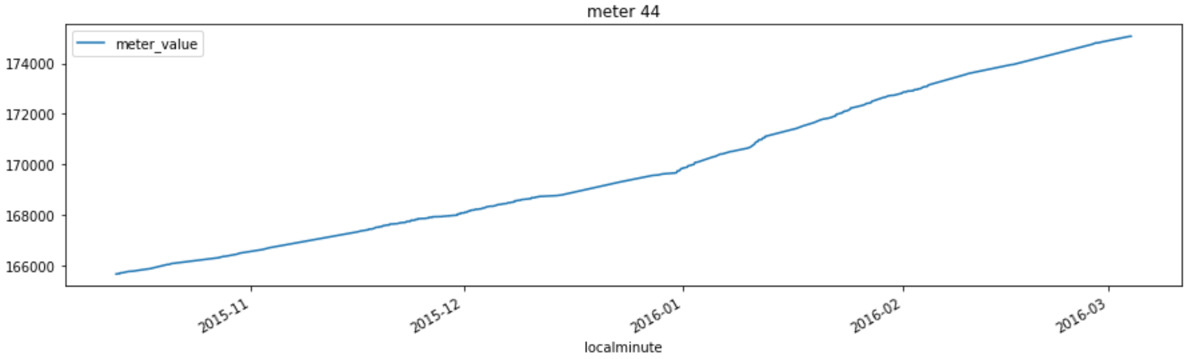
Print full-length (6 mth) plot by meterid.

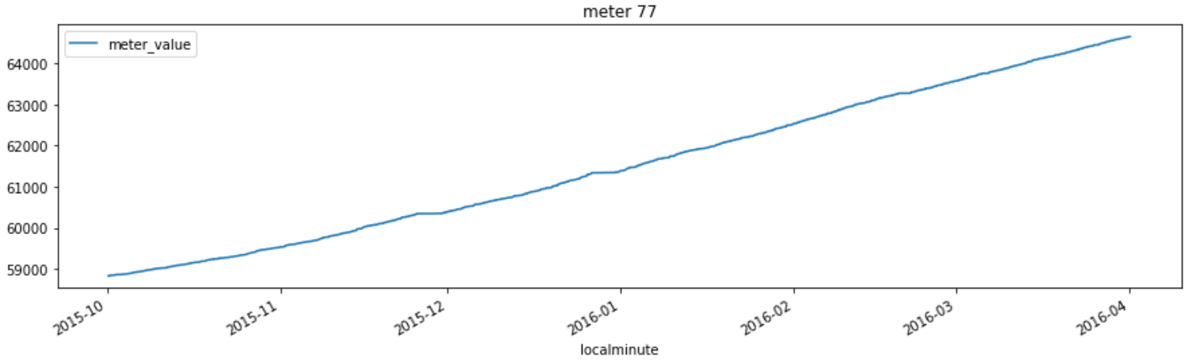
In [35]:

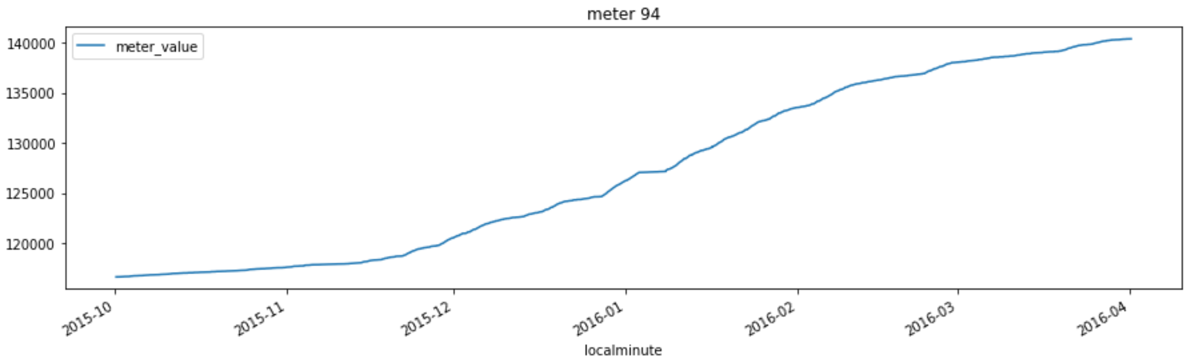
```
for key in keys_list:
    df_i = groups.get_group(key)
    df_i.drop(columns='dataid').plot(figsize=(15,4), title=str(f'meter {key}'))
```

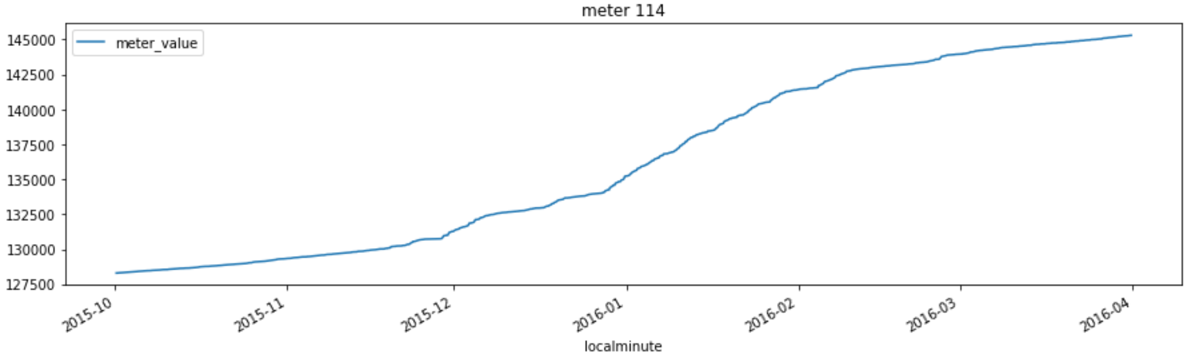
```
C:\Users\Melvin\Anaconda3\lib\site-packages\matplotlib\pyplot.py:537: Runtime
Warning: More than 20 figures have been opened. Figures created through the p
yplot interface (`matplotlib.pyplot.figure`) are retained until explicitly cl
osed and may consume too much memory. (To control this warning, see the rcPar
am `figure.max_open_warning`).
  max_open_warning, RuntimeWarning)
```

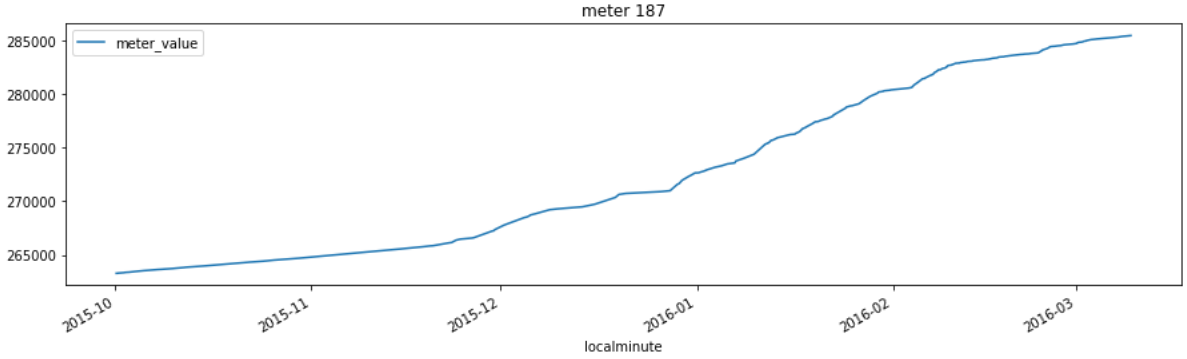


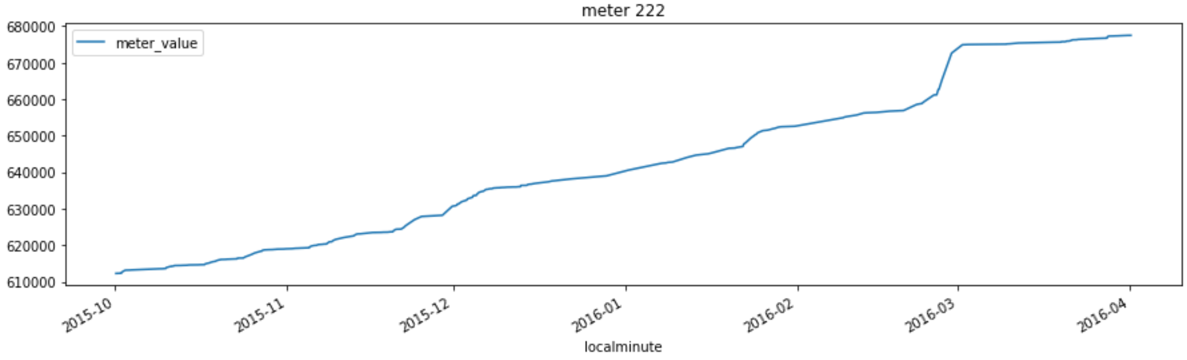


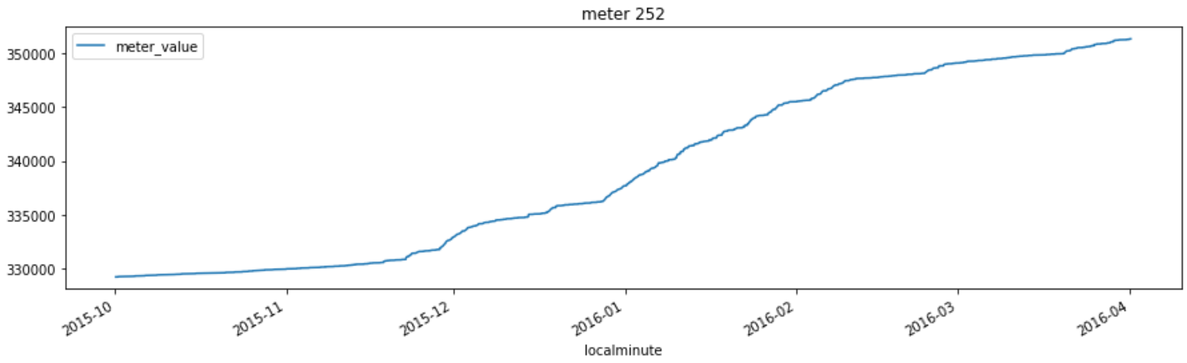


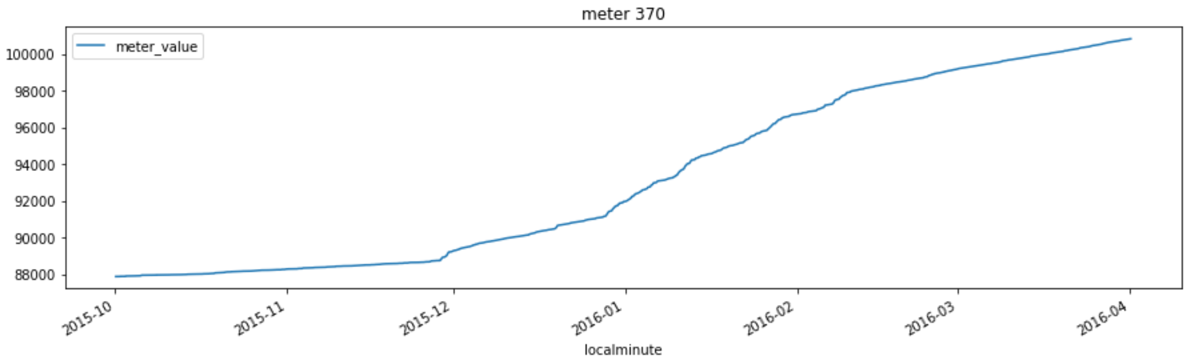

meter 94
meter_value
140000
135000
130000
125000
120000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

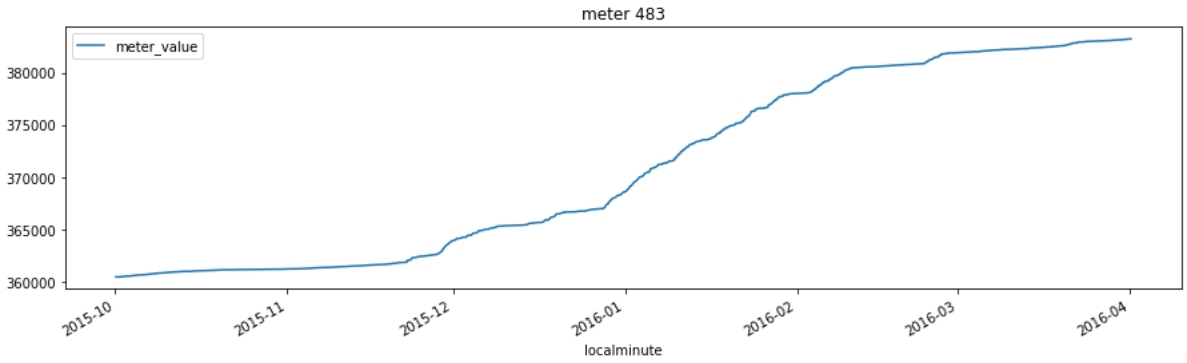

meter 114
meter_value
145000
142500
140000
137500
135000
132500
130000
127500
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

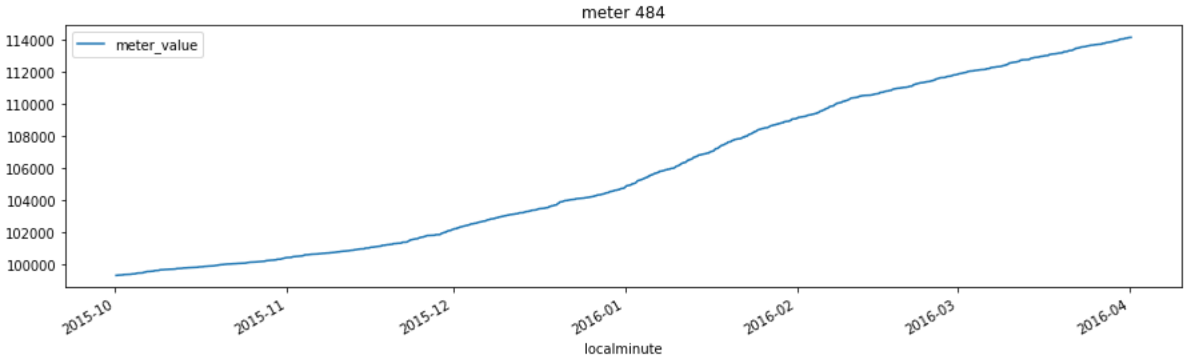

meter 187
meter_value
285000
280000
275000
270000
265000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
localminute

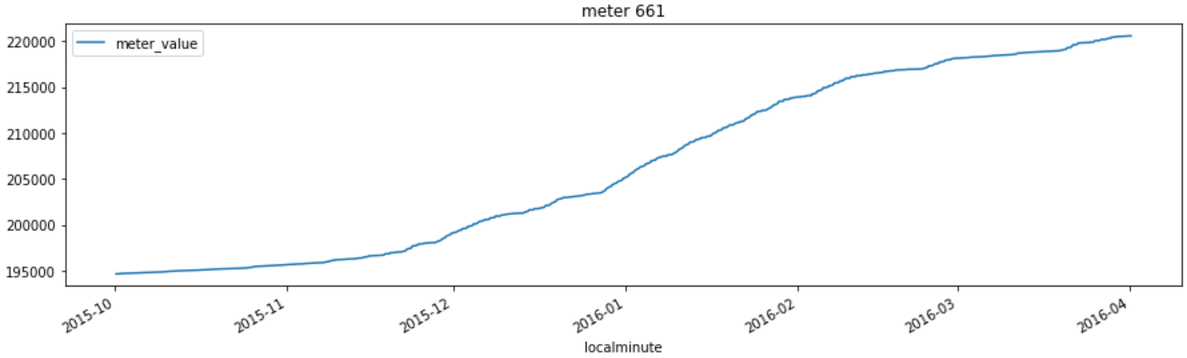

meter 222
meter_value
680000
670000
660000
650000
640000
630000
620000
610000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

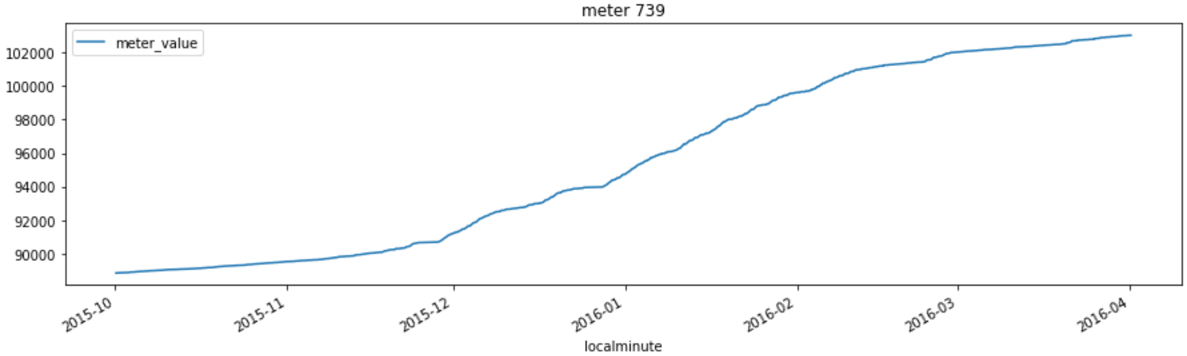

meter 252
meter_value
350000
345000
340000
335000
330000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

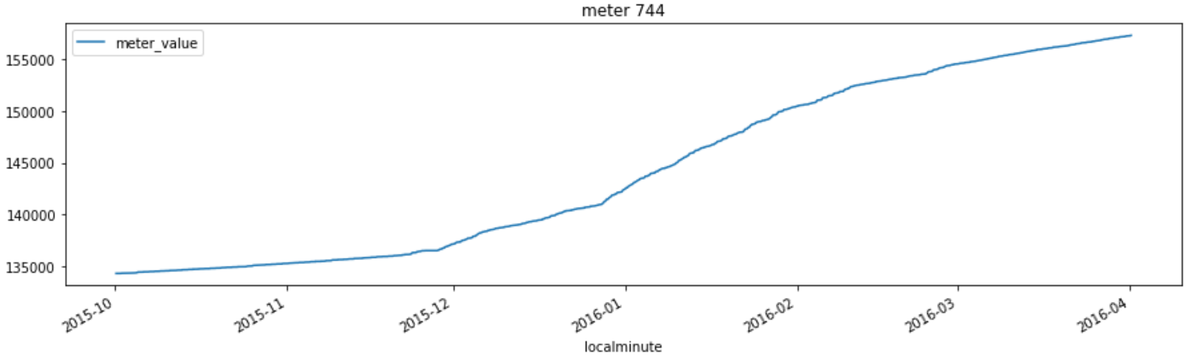

meter 370
meter_value
100000
98000
96000
94000
92000
90000
88000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

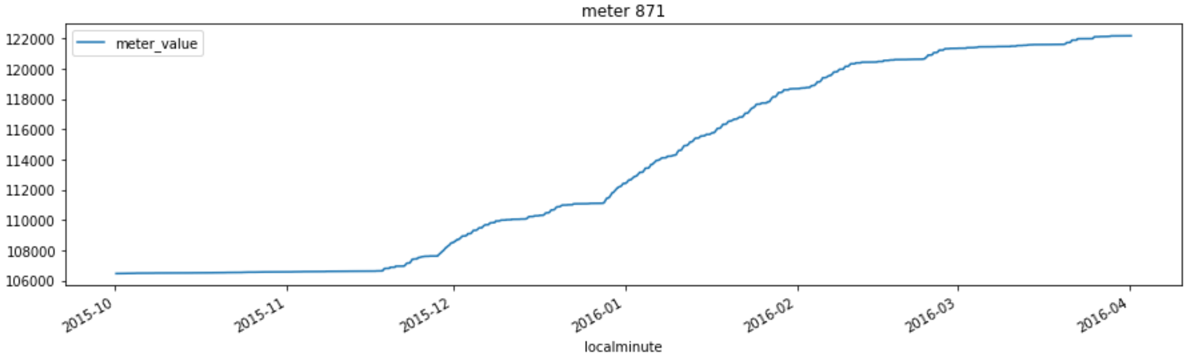

meter 483
meter_value
380000
375000
370000
365000
360000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

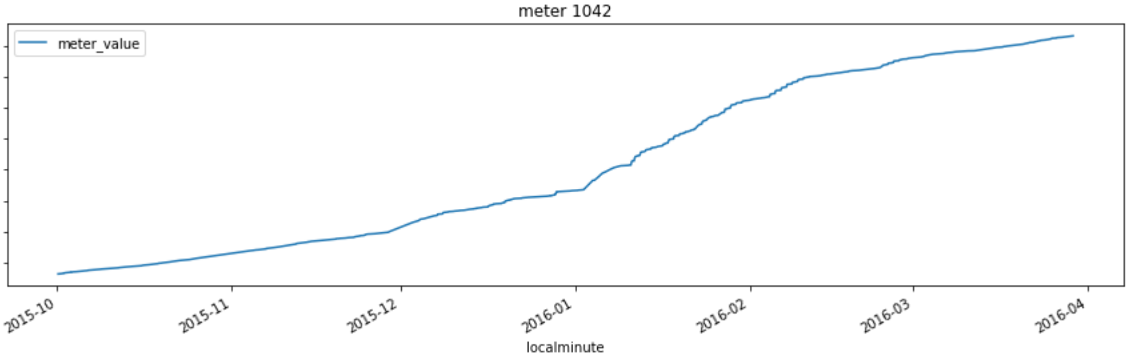

meter 484
meter_value
114000
112000
110000
108000
106000
104000
102000
100000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

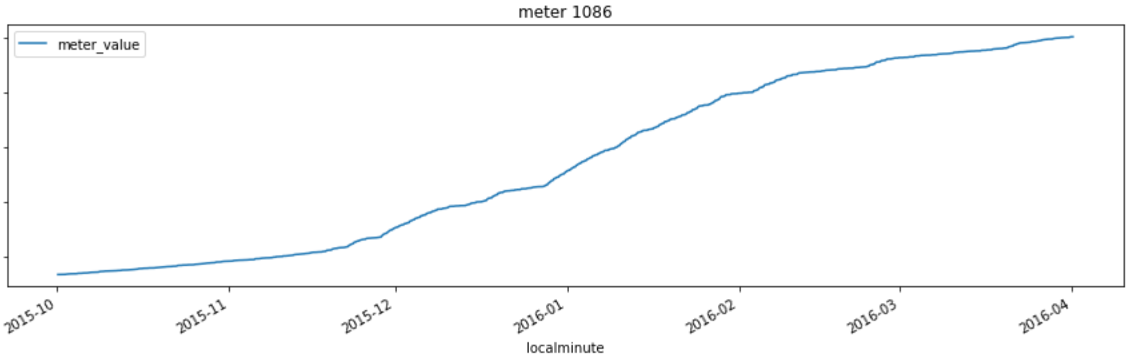

meter 661

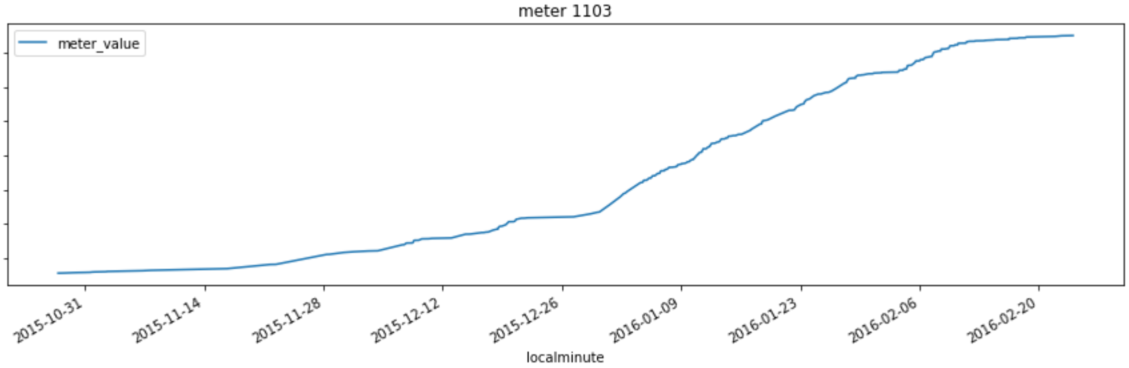

meter 739

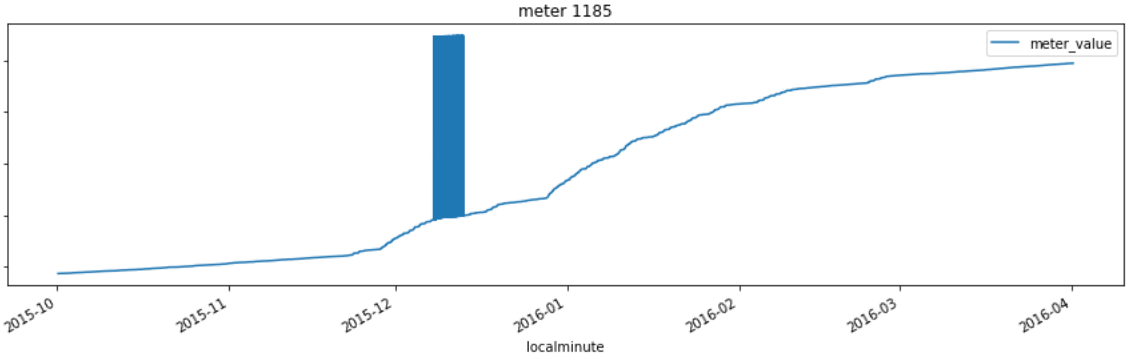

meter 744

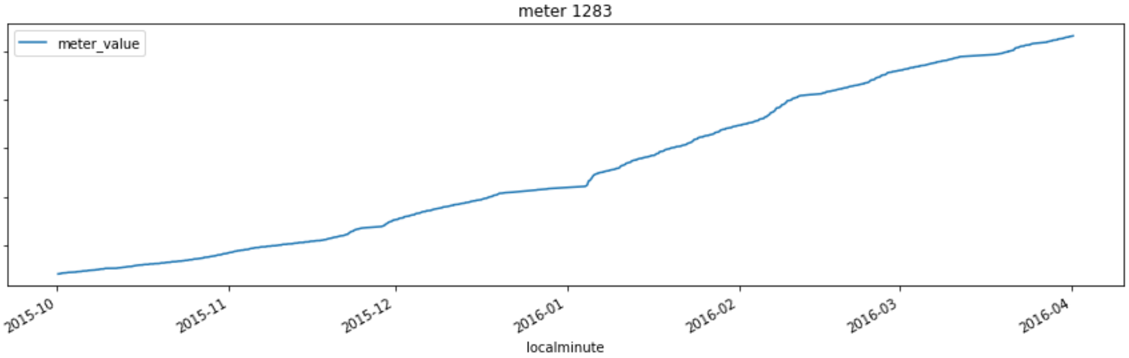

meter 871

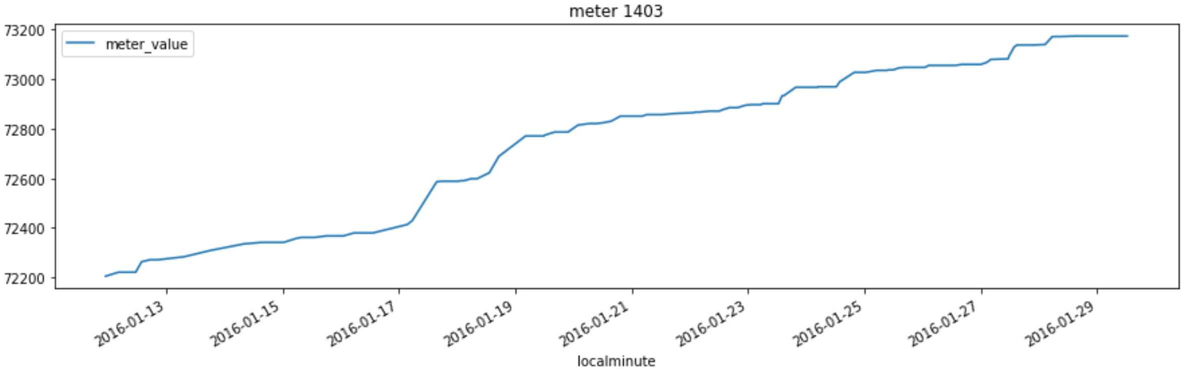

meter 1042
meter_value
218000
216000
214000
212000
210000
208000
206000
204000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

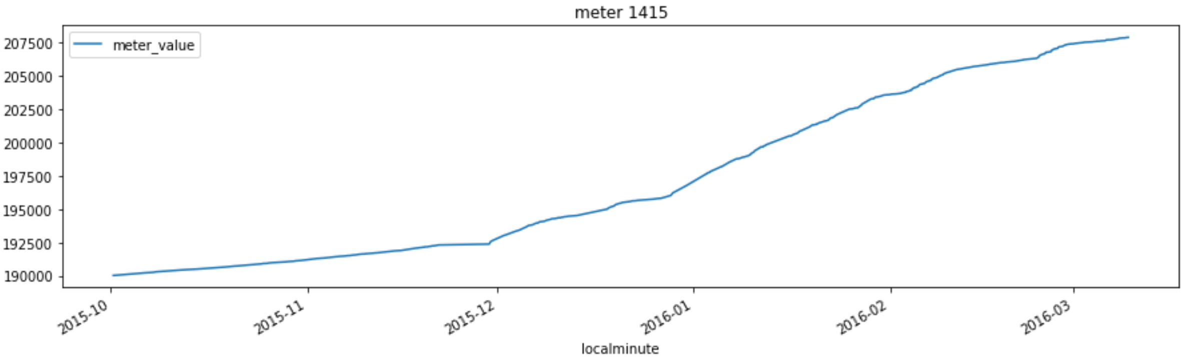

meter 1086
meter_value
105000
100000
95000
90000
85000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

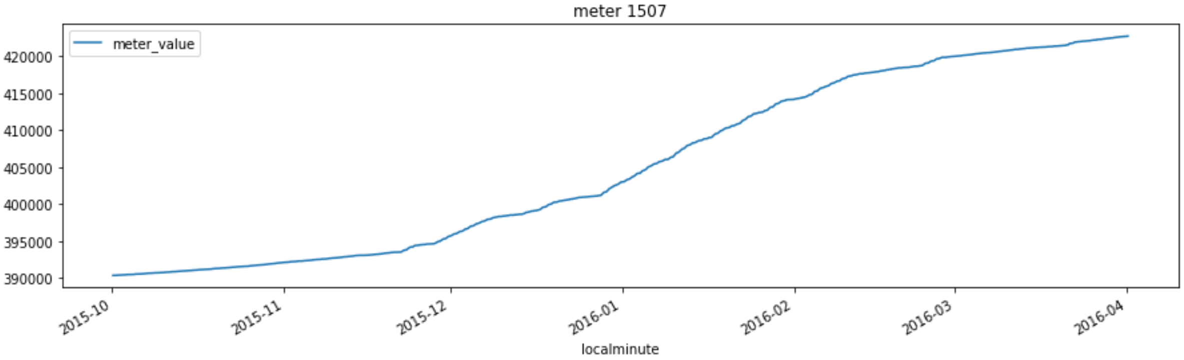

meter 1103
meter_value
196000
194000
192000
190000
188000
186000
184000
2015-10-31
2015-11-14
2015-11-28
2015-12-12
2015-12-26
2016-01-09
2016-01-23
2016-02-06
2016-02-20
localminute

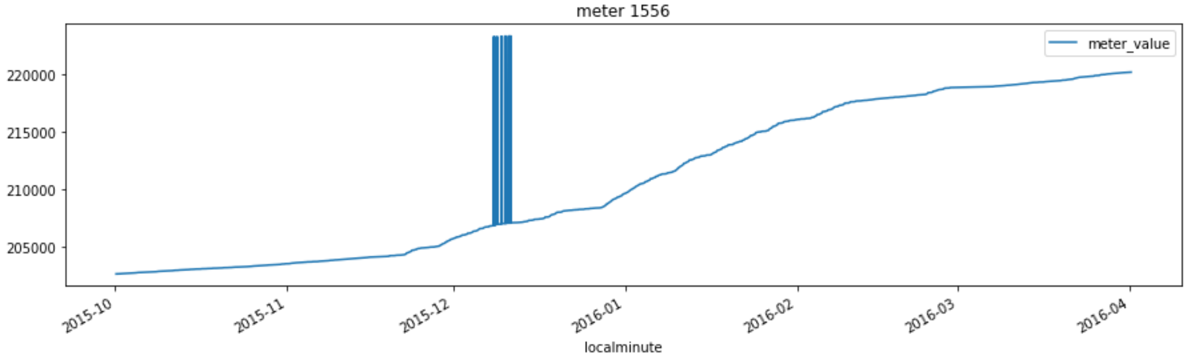

meter 1185
meter_value
160000
155000
150000
145000
140000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

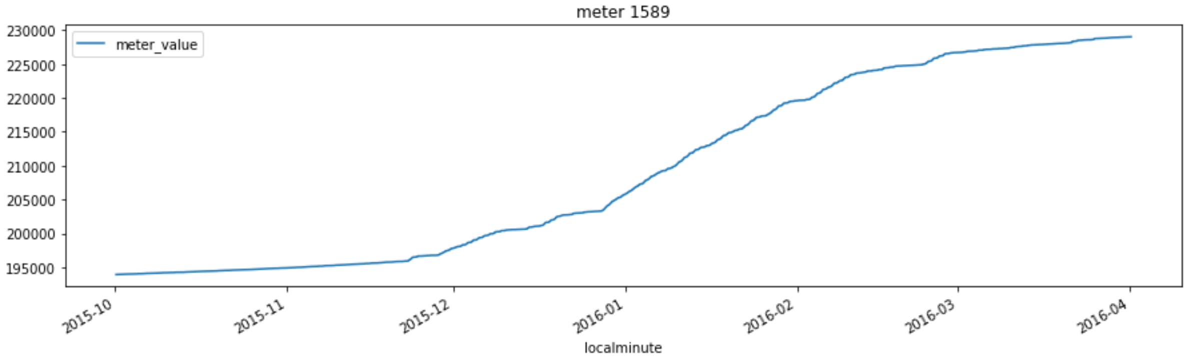

meter 1283
meter_value
190000
185000
180000
175000
170000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

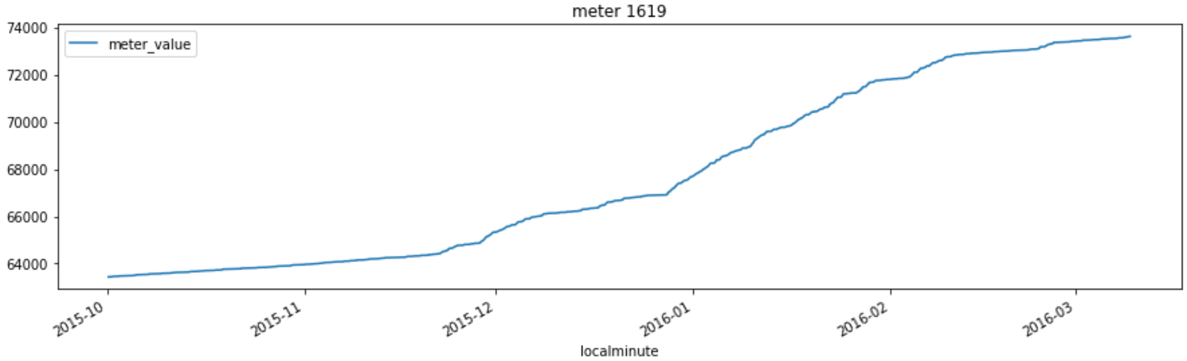

meter 1403
meter_value
73200
73000
72800
72600
72400
72200
2016-01-13
2016-01-15
2016-01-17
2016-01-19
2016-01-21
2016-01-23
2016-01-25
2016-01-27
2016-01-29
localminute

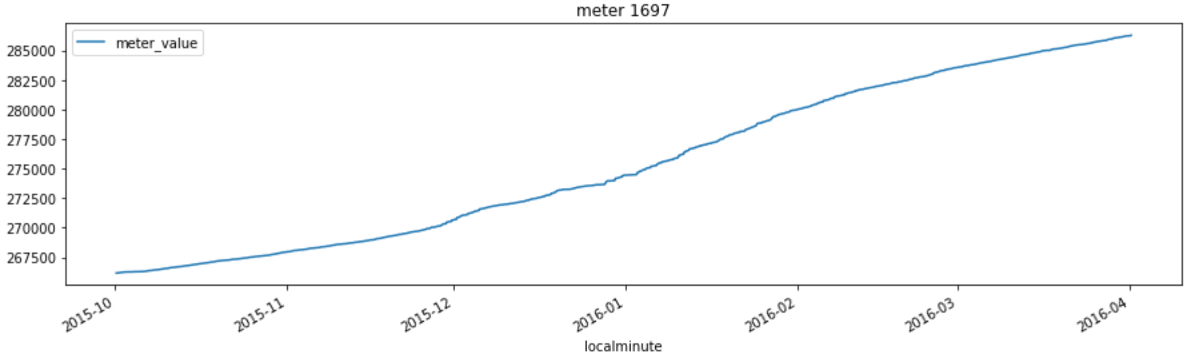

meter 1415
meter_value
207500
205000
202500
200000
197500
195000
192500
190000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
localminute

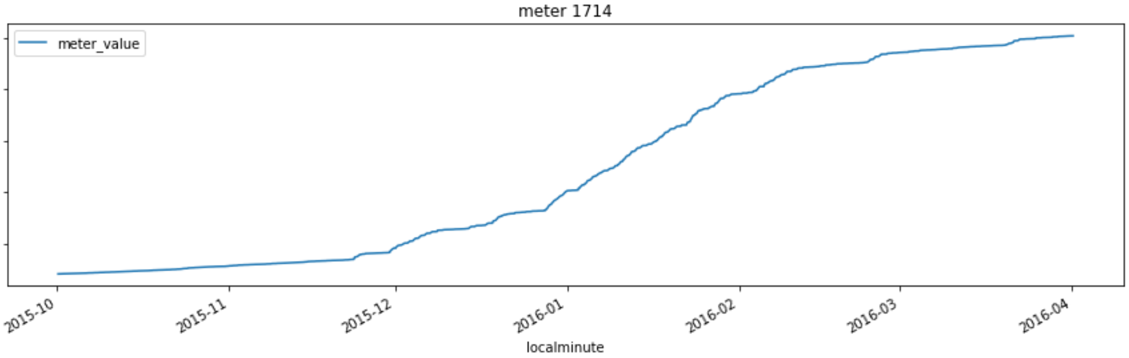

meter 1507
meter_value
420000
415000
410000
405000
400000
395000
390000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

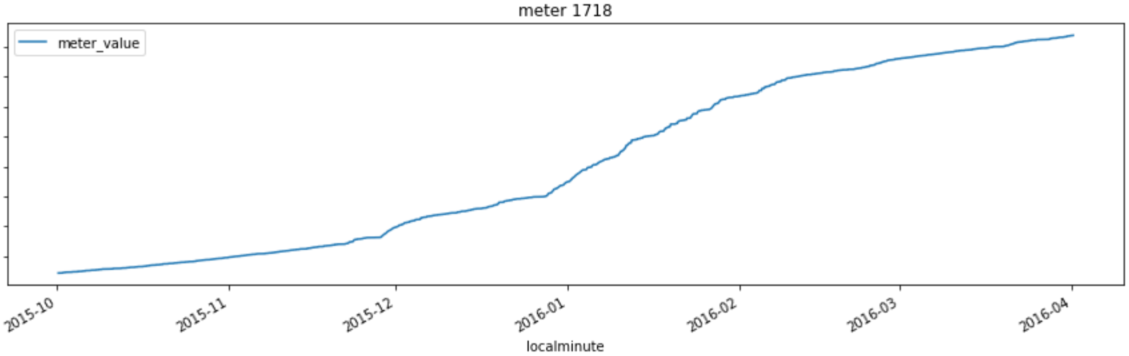

meter 1556
meter_value
220000
215000
210000
205000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

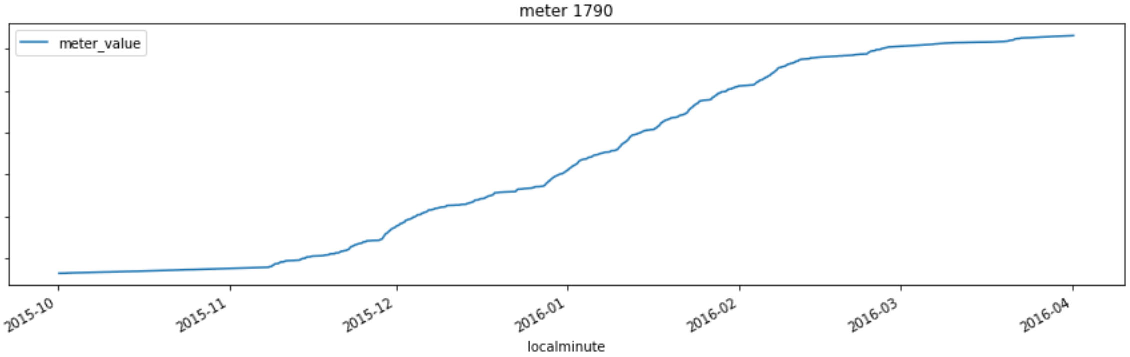

meter 1589
meter_value
230000
225000
220000
215000
210000
205000
200000
195000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

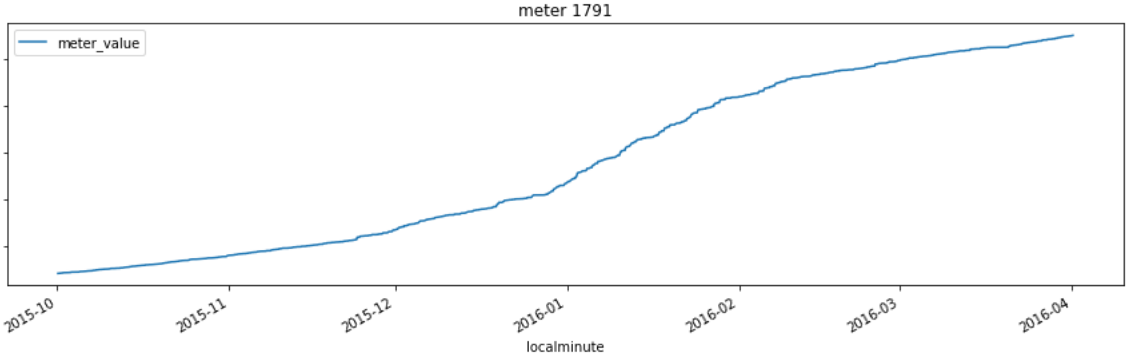

meter 1619
meter_value
74000
72000
70000
68000
66000
64000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
localminute

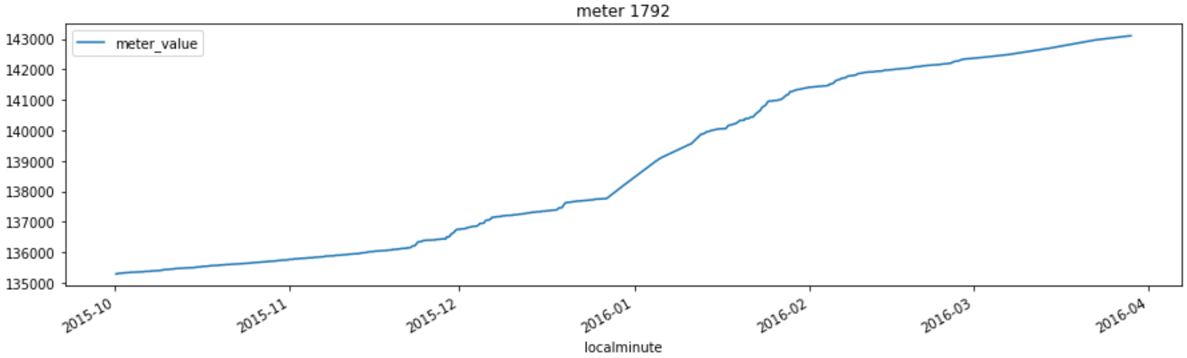

meter 1697
meter_value
285000
282500
280000
277500
275000
272500
270000
267500
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

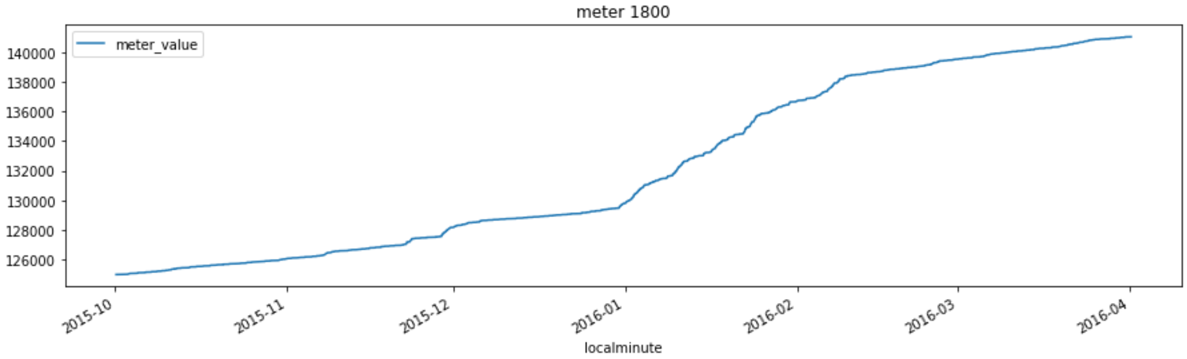

meter 1714
meter_value
170000
165000
160000
155000
150000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

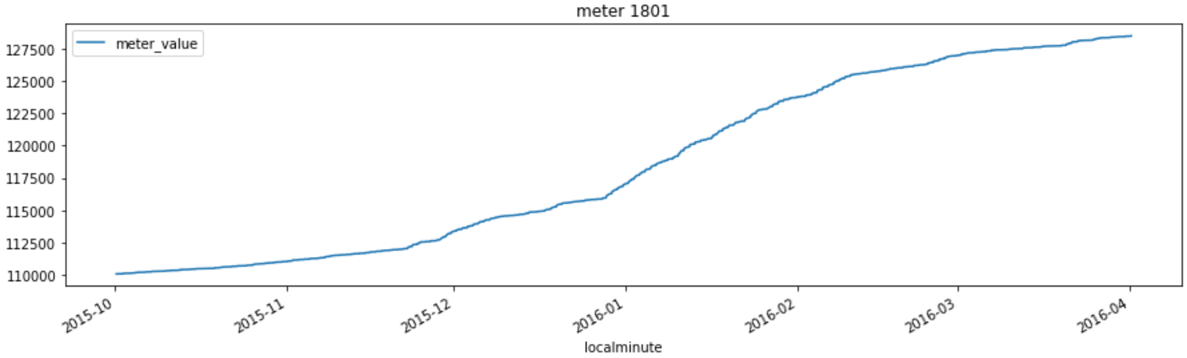

meter 1718
meter_value
180000
177500
175000
172500
170000
167500
165000
162500
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

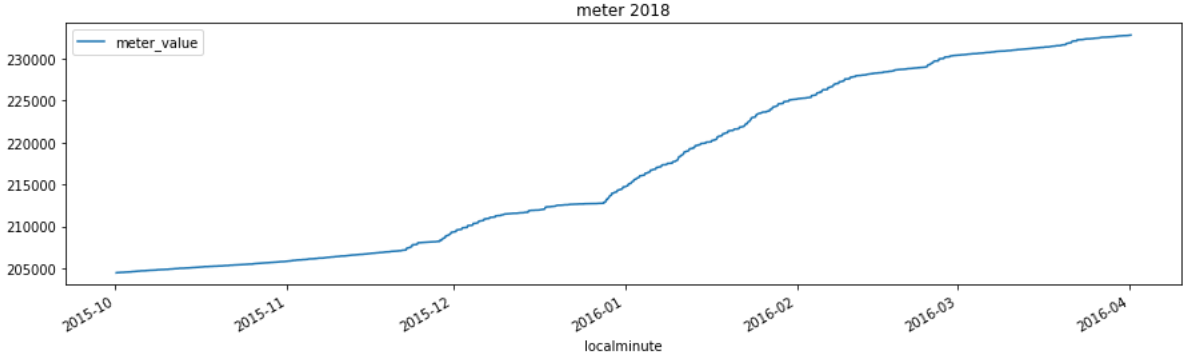

meter 1790
meter_value
350000
340000
330000
320000
310000
300000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

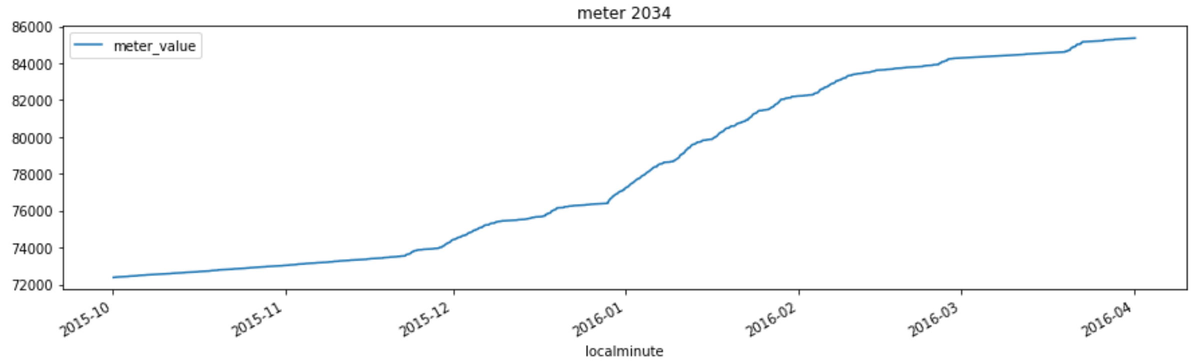

meter 1791
meter_value
140000
138000
136000
134000
132000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

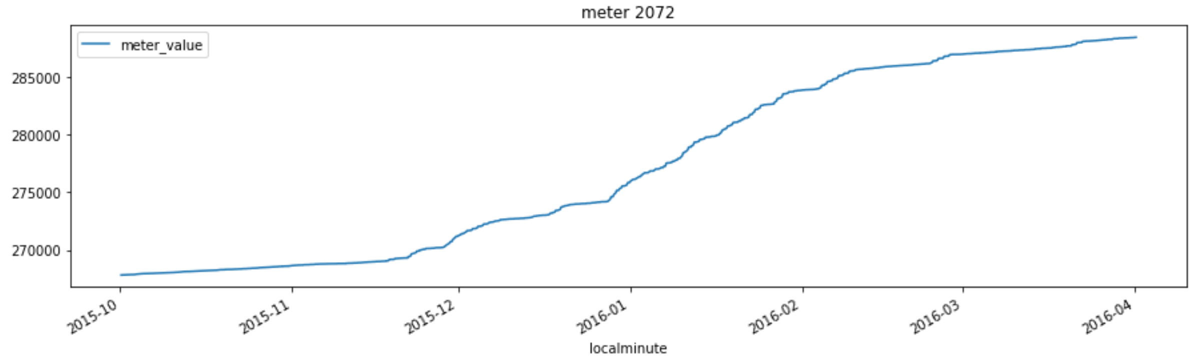

meter 1792
meter_value
143000
142000
141000
140000
139000
138000
137000
136000
135000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

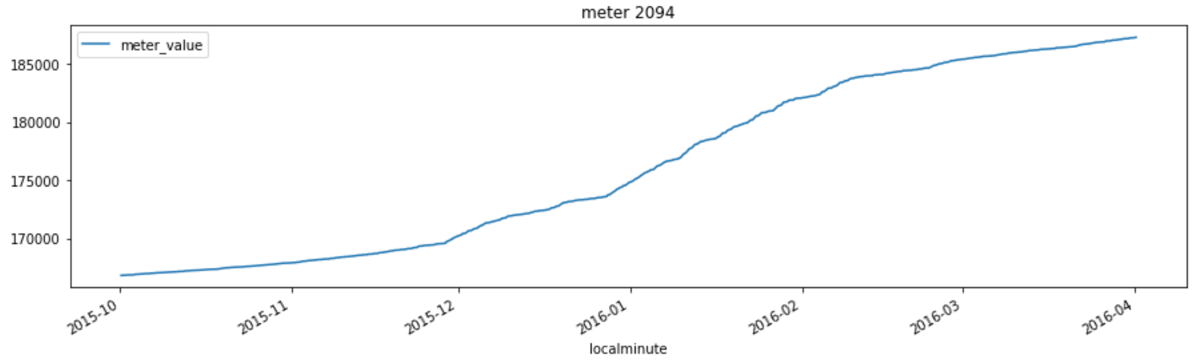

meter 1800
meter_value
140000
138000
136000
134000
132000
130000
128000
126000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

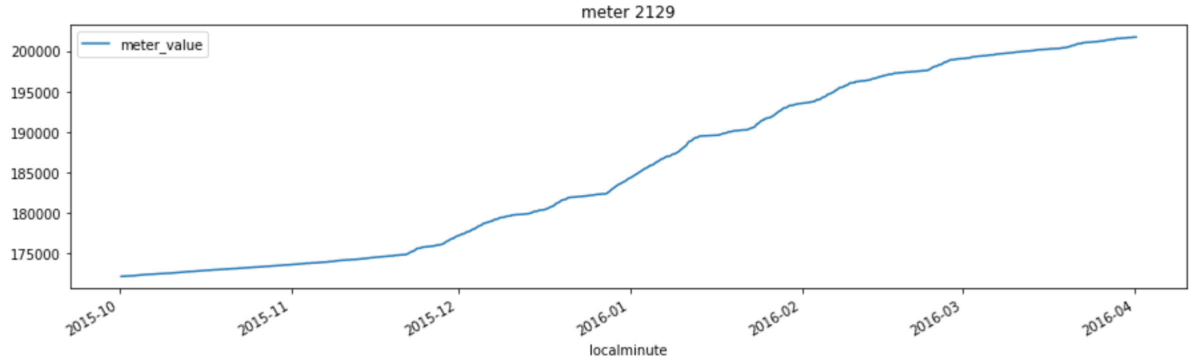

meter 1801
meter_value
127500
125000
122500
120000
117500
115000
112500
110000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

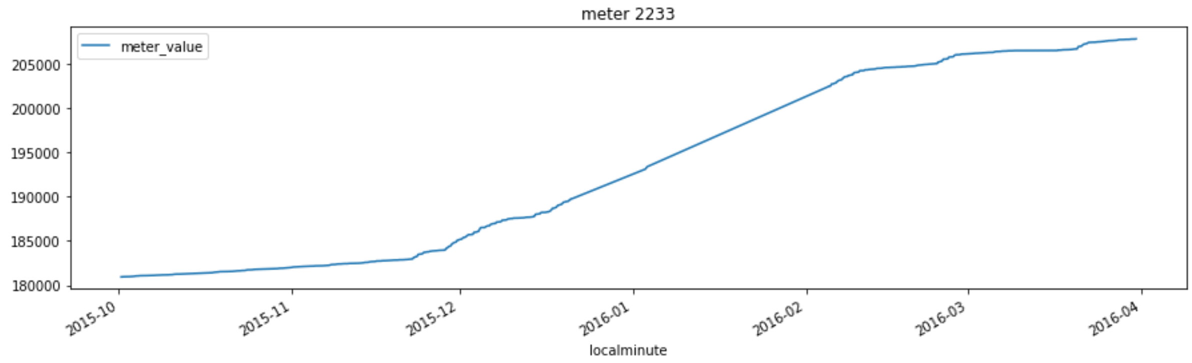

meter 2018
meter_value
230000
225000
220000
215000
210000
205000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

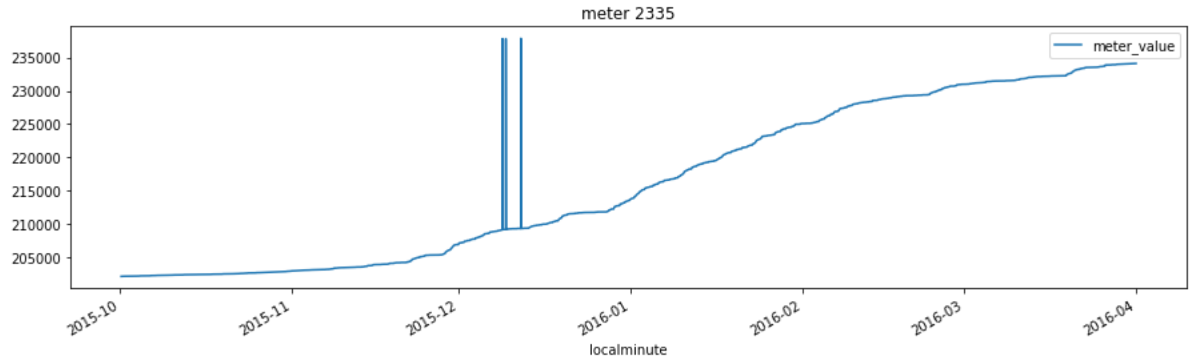

meter 2034
meter_value
86000
84000
82000
80000
78000
76000
74000
72000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

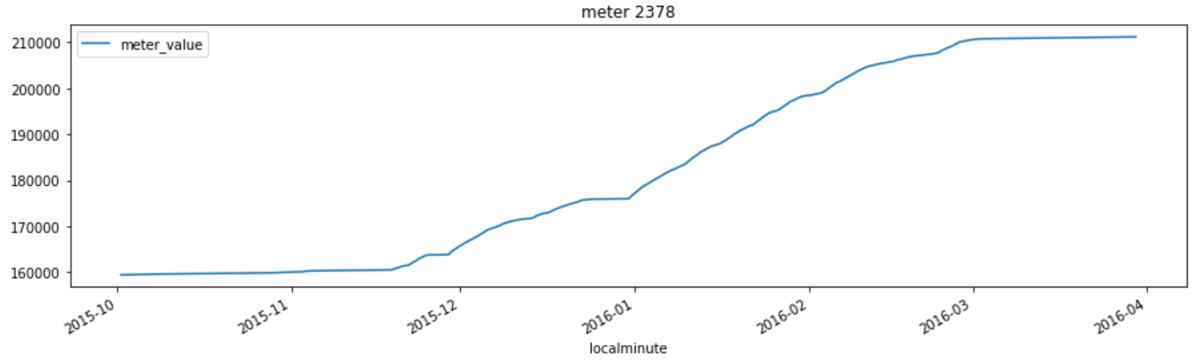

meter 2072
meter_value
285000
280000
275000
270000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

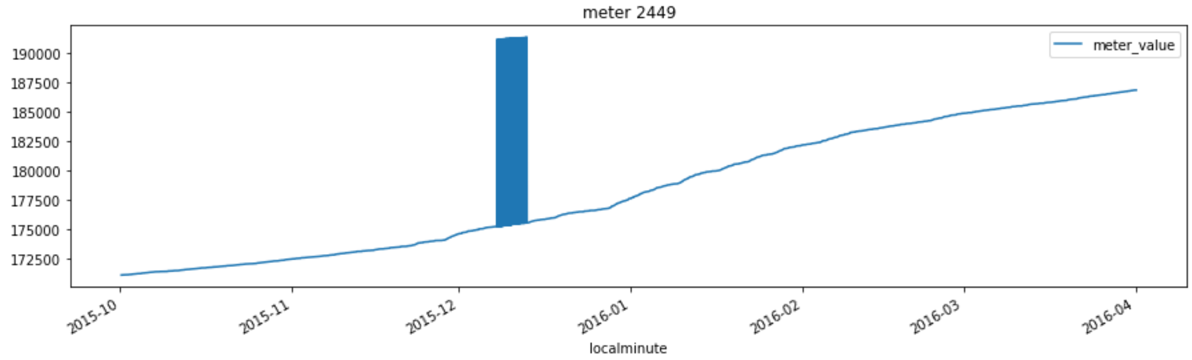

meter 2094
meter_value
185000
180000
175000
170000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

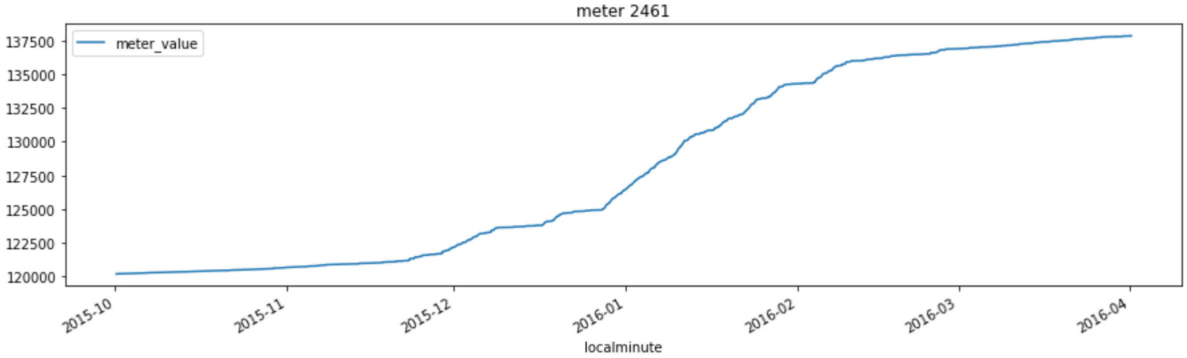

meter 2129
meter_value
200000
195000
190000
185000
180000
175000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

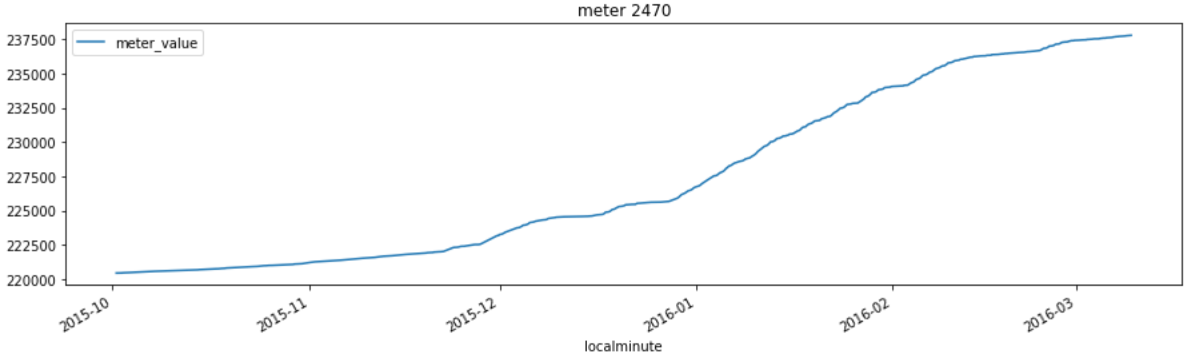

meter 2233
meter_value
205000
200000
195000
190000
185000
180000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

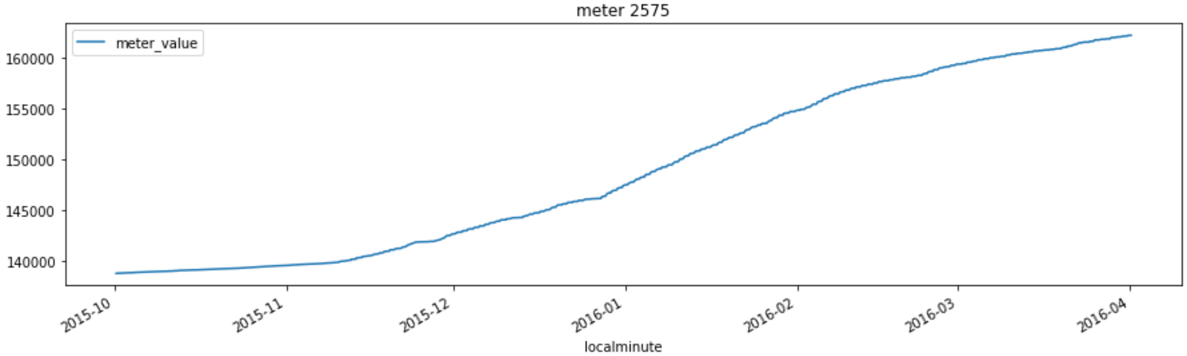

meter 2335
meter_value
235000
230000
225000
220000
215000
210000
205000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

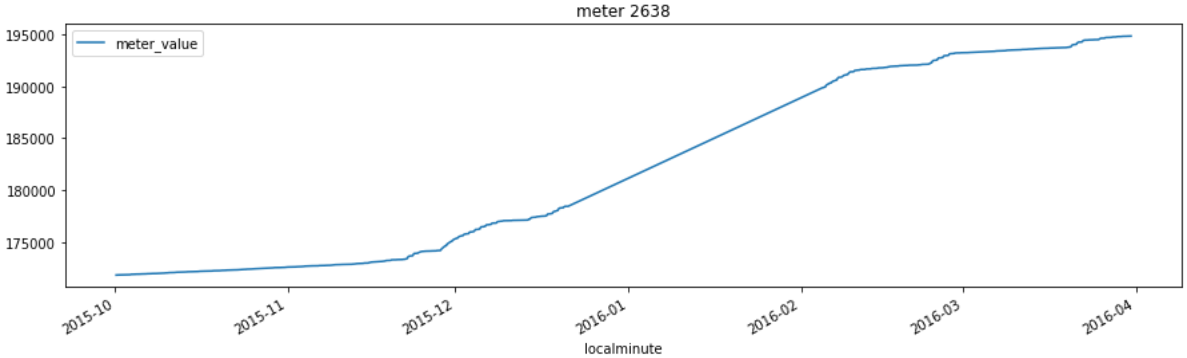

meter 2378
meter_value
210000
200000
190000
180000
170000
160000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

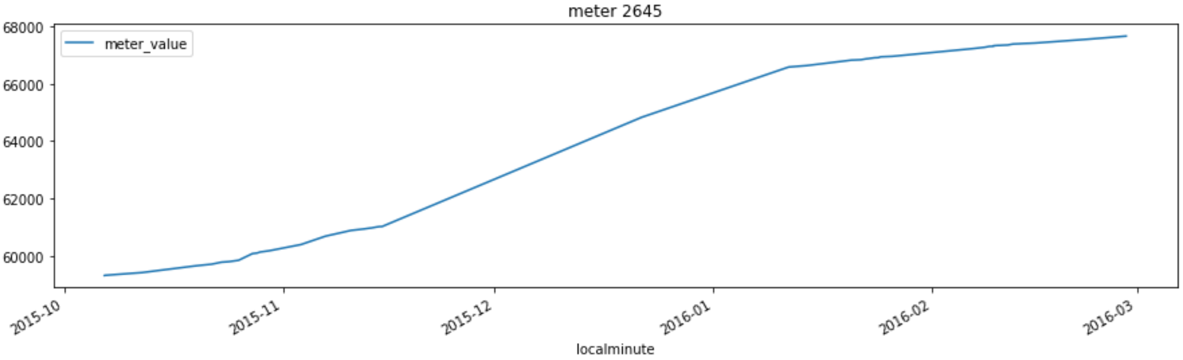

meter 2449
meter_value
190000
187500
185000
182500
180000
177500
175000
172500
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

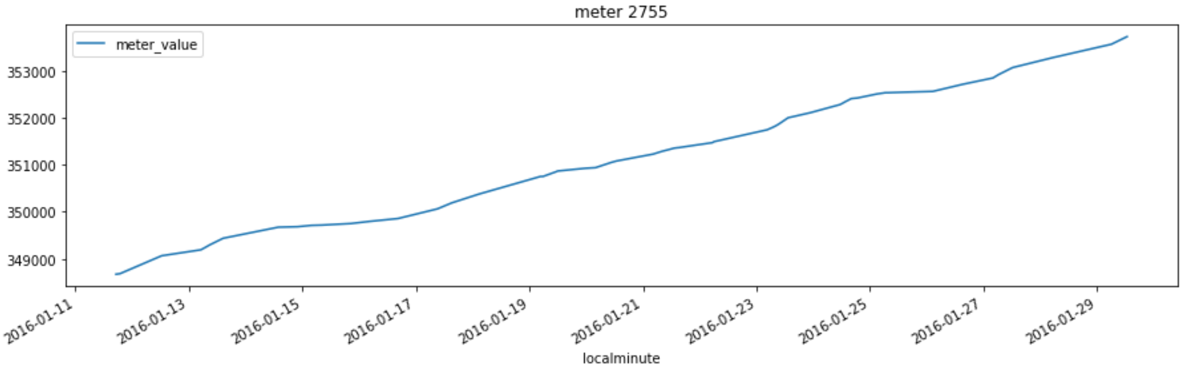

meter 2461
meter_value
137500
135000
132500
130000
127500
125000
122500
120000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

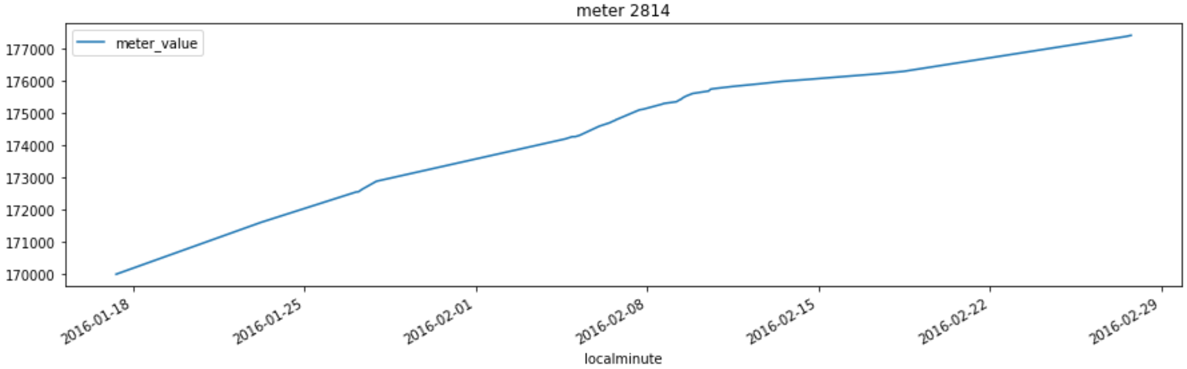

meter 2470
meter_value
237500
235000
232500
230000
227500
225000
222500
220000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
localminute

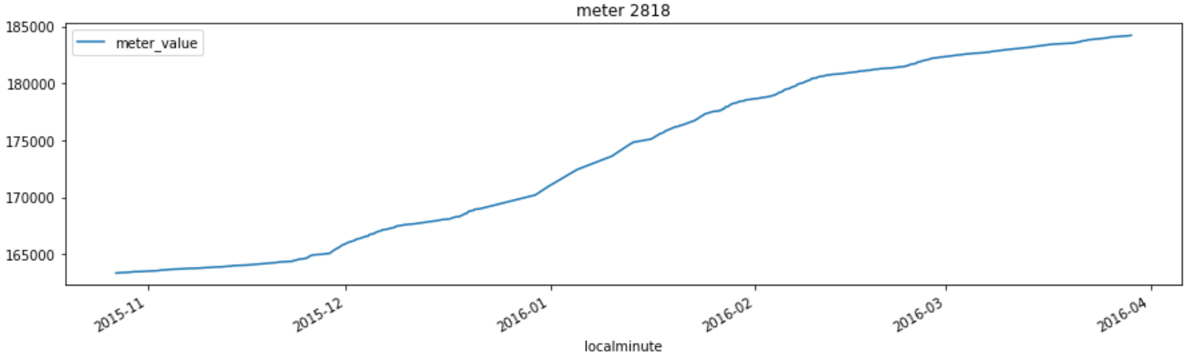

meter 2575
meter_value
160000
155000
150000
145000
140000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

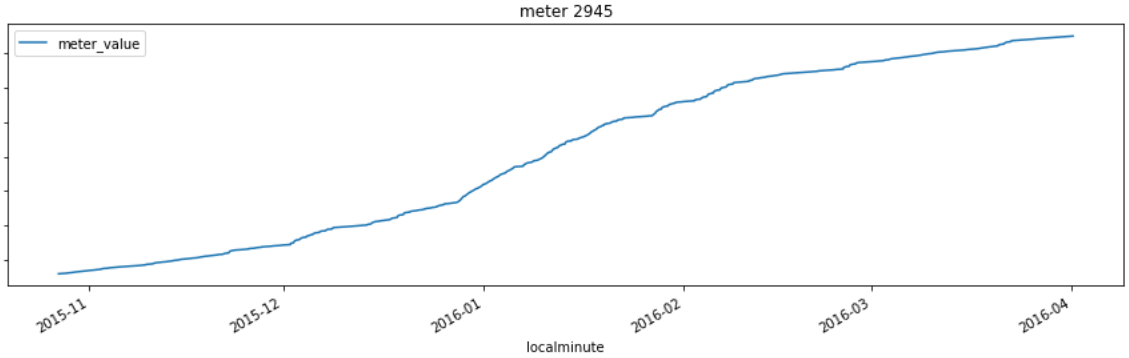

meter 2638
meter_value
195000
190000
185000
180000
175000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

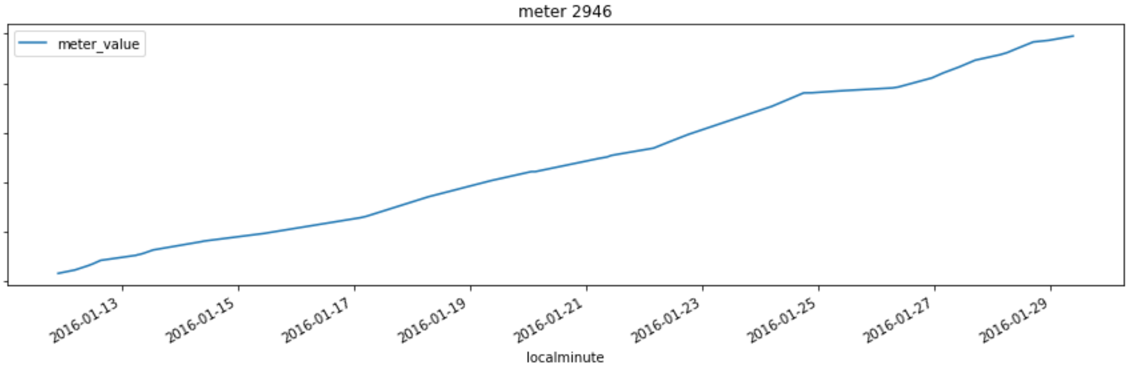

meter 2645
meter_value
68000
66000
64000
62000
60000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
localminute

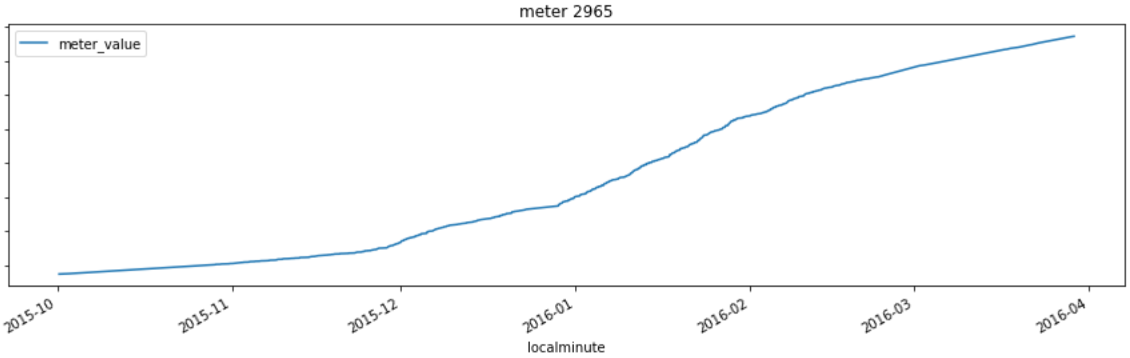

meter 2755
meter_value
353000
352000
351000
350000
349000
2016-01-11
2016-01-13
2016-01-15
2016-01-17
2016-01-19
2016-01-21
2016-01-23
2016-01-25
2016-01-27
2016-01-29
localminute

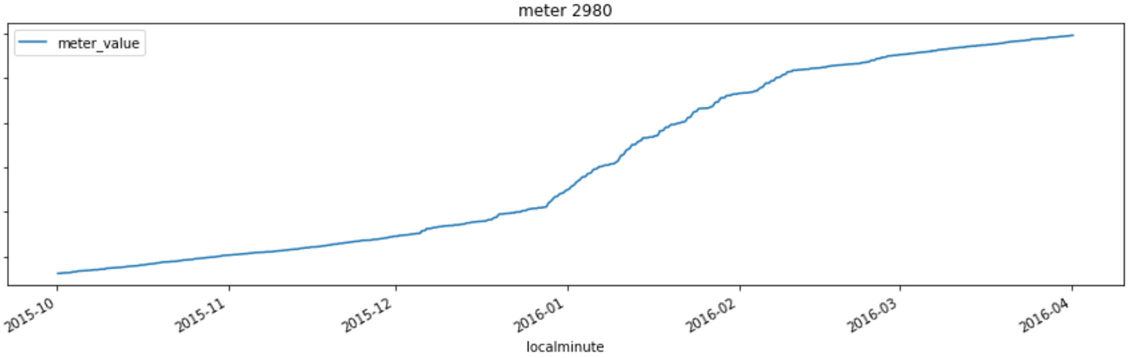

meter 2814
meter_value
177000
176000
175000
174000
173000
172000
171000
170000
2016-01-18
2016-01-25
2016-02-01
2016-02-08
2016-02-15
2016-02-22
2016-02-29
localminute

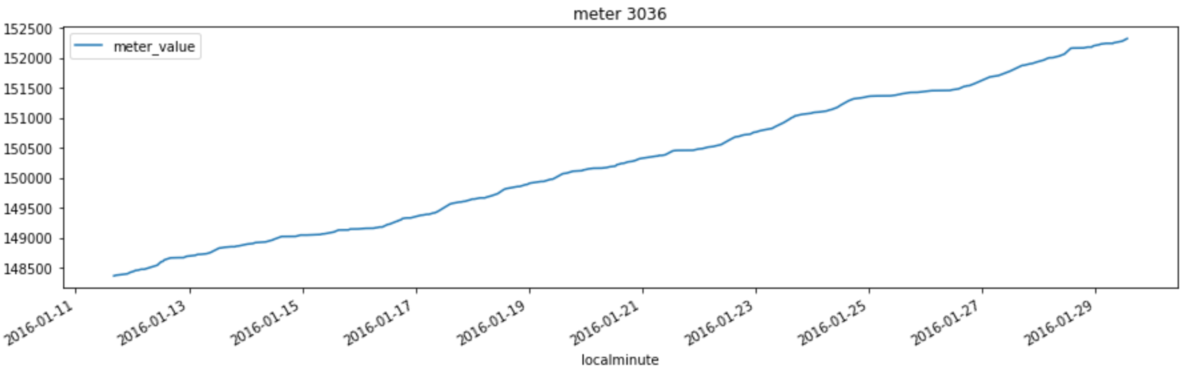

meter 2818
meter_value
185000
180000
175000
170000
165000
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

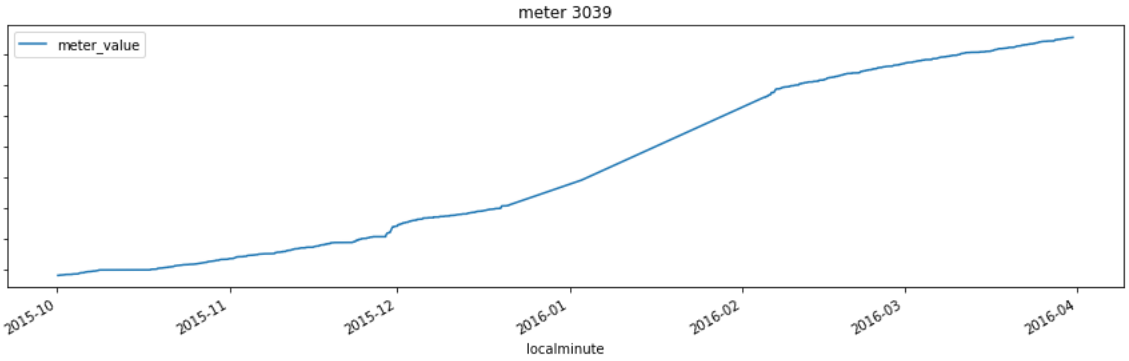

meter 2945
meter_value
160000
157500
155000
152500
150000
147500
145000
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

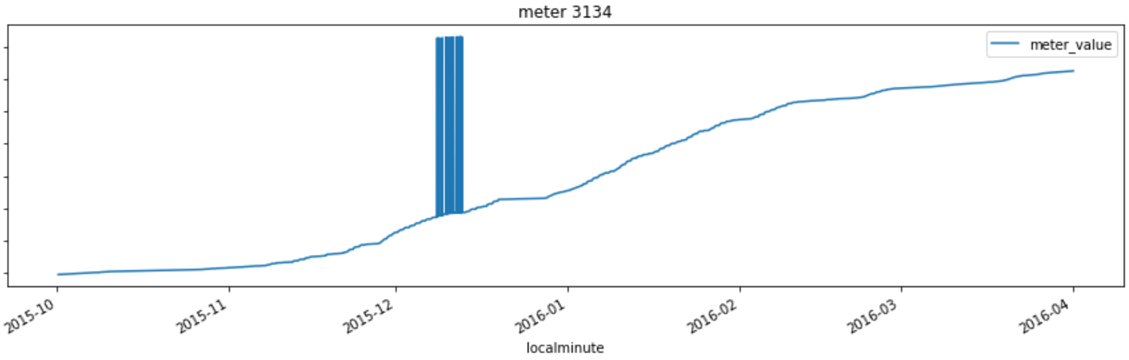

meter 2946
meter_value
161000
160000
159000
158000
157000
156000
2016-01-13
2016-01-15
2016-01-17
2016-01-19
2016-01-21
2016-01-23
2016-01-25
2016-01-27
2016-01-29
localminute

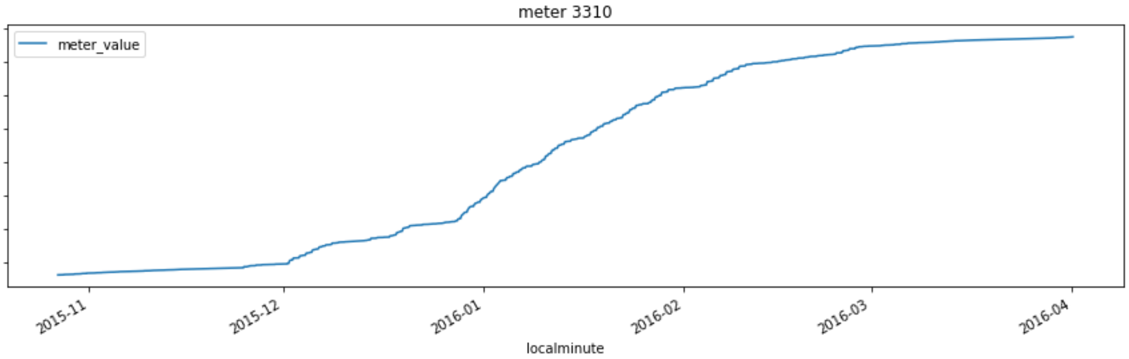

meter 2965
meter_value
206000
204000
202000
200000
198000
196000
194000
192000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

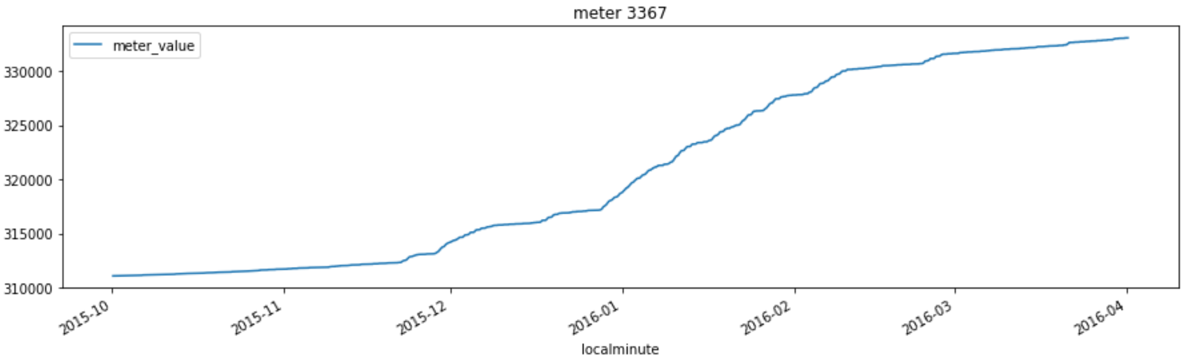

meter 2980
meter_value
146000
144000
142000
140000
138000
136000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

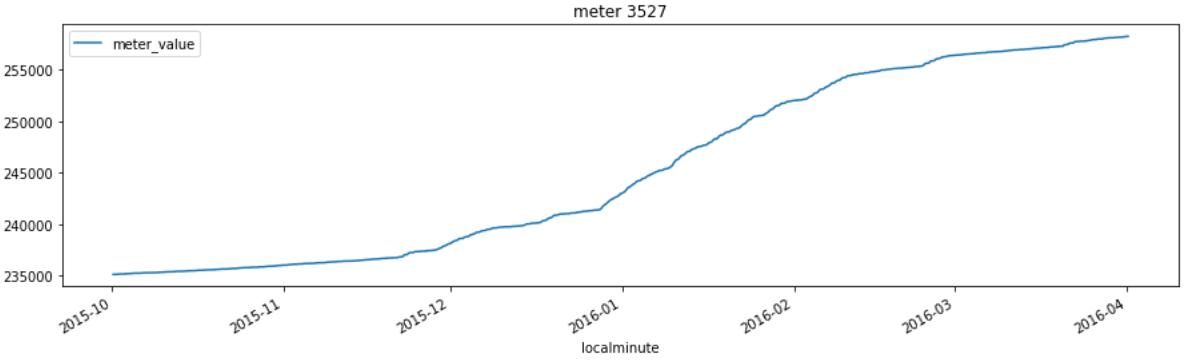

meter 3036
meter_value
152500
152000
151500
151000
150500
150000
149500
149000
148500
2016-01-11
2016-01-13
2016-01-15
2016-01-17
2016-01-19
2016-01-21
2016-01-23
2016-01-25
2016-01-27
2016-01-29
localminute

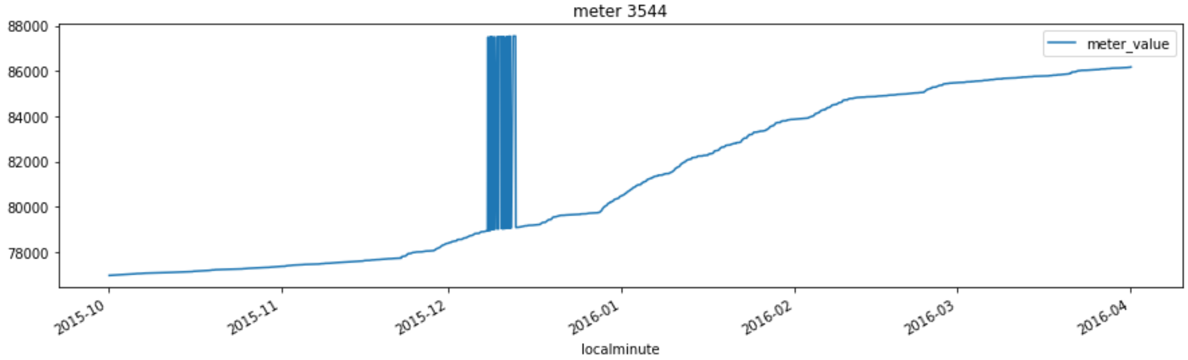

meter 3039
meter_value
148000
146000
144000
142000
140000
138000
136000
134000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

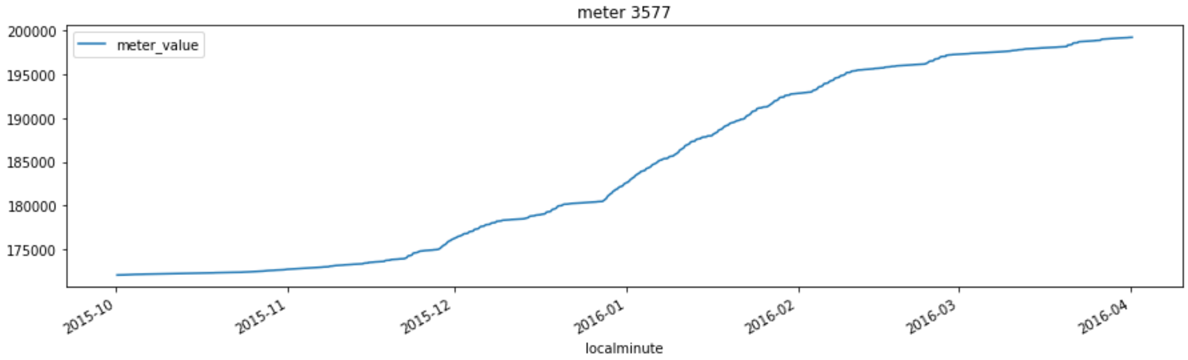

meter 3134
meter_value
180000
177500
175000
172500
170000
167500
165000
162500
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

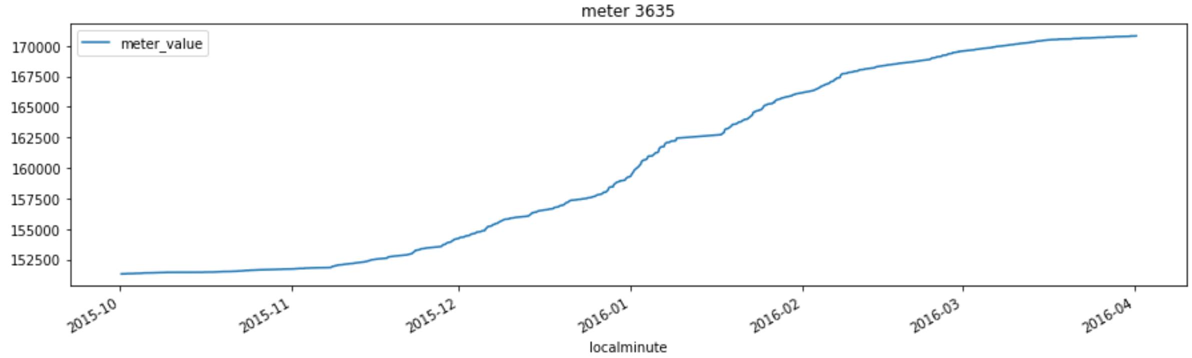

meter 3310
meter_value
406000
404000
402000
400000
398000
396000
394000
392000
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

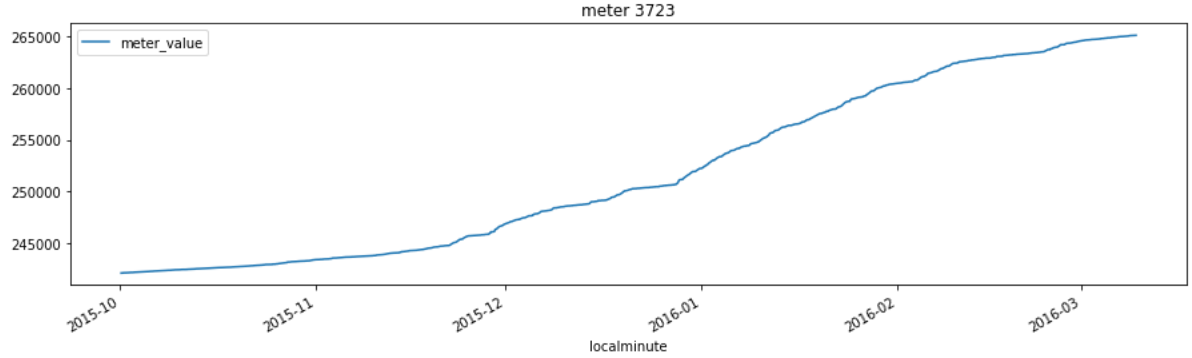

meter 3367
meter_value
330000
325000
320000
315000
310000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

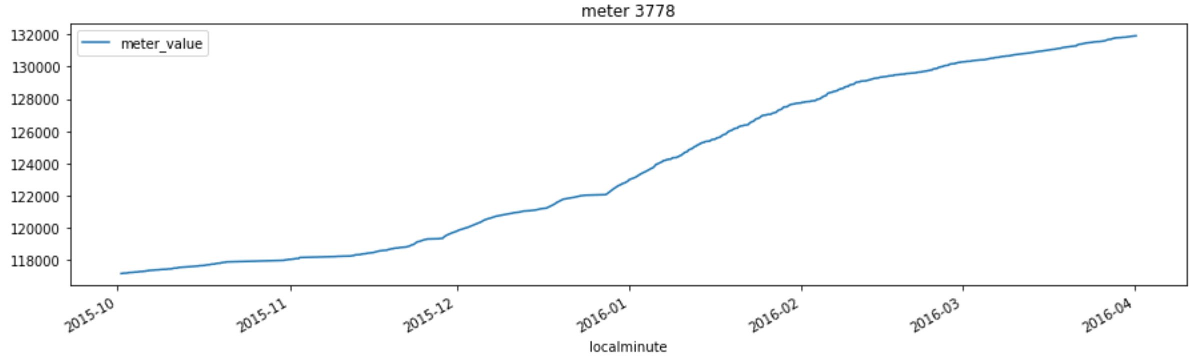

meter 3527
meter_value
255000
250000
245000
240000
235000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

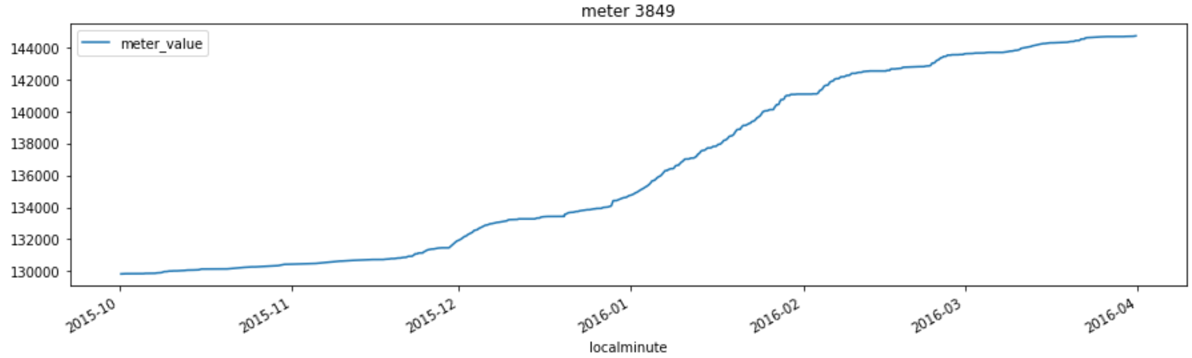

meter 3544
meter_value
88000
86000
84000
82000
80000
78000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

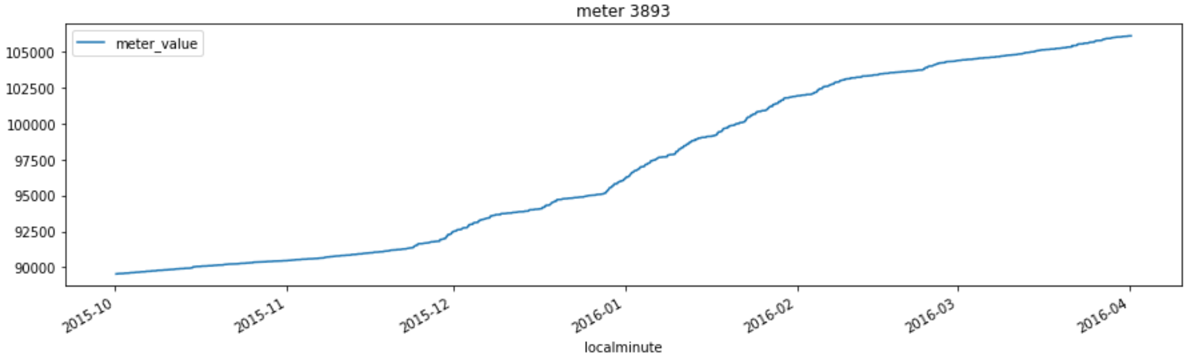

meter 3577
meter_value
200000
195000
190000
185000
180000
175000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

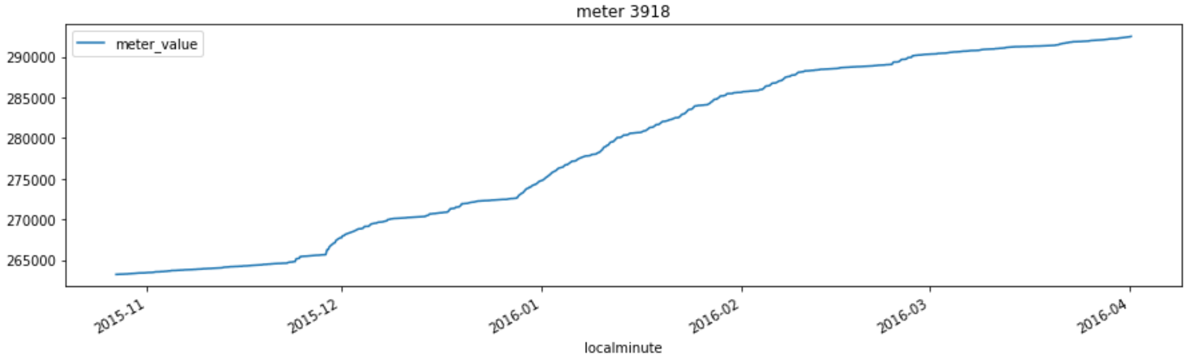

meter 3635
meter_value
170000
167500
165000
162500
160000
157500
155000
152500
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

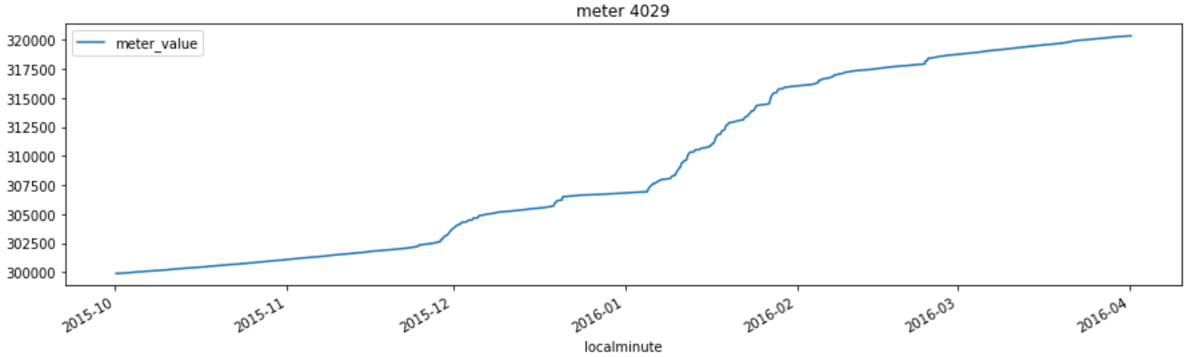

meter 3723
meter_value
265000
260000
255000
250000
245000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
localminute

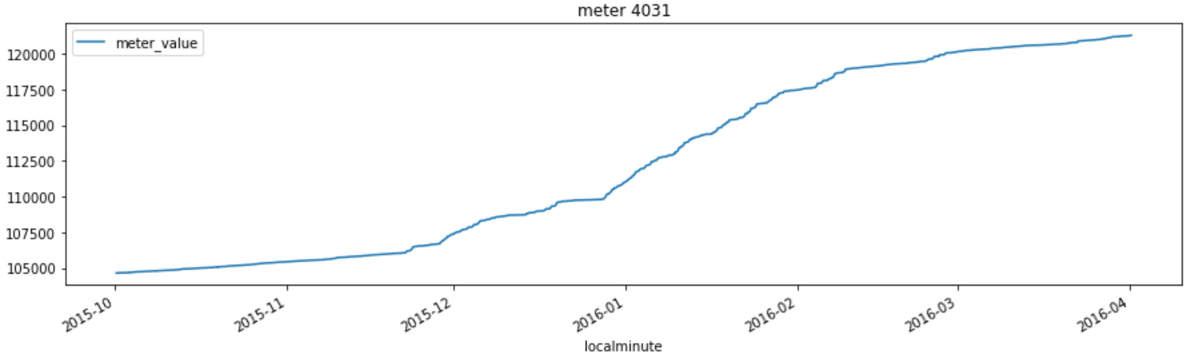

meter 3778
meter_value
132000
130000
128000
126000
124000
122000
120000
118000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

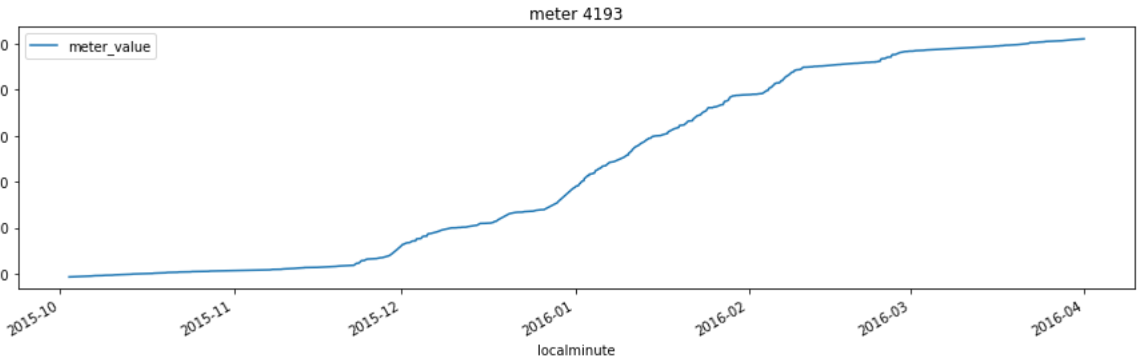

meter 3849
meter_value
144000
142000
140000
138000
136000
134000
132000
130000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

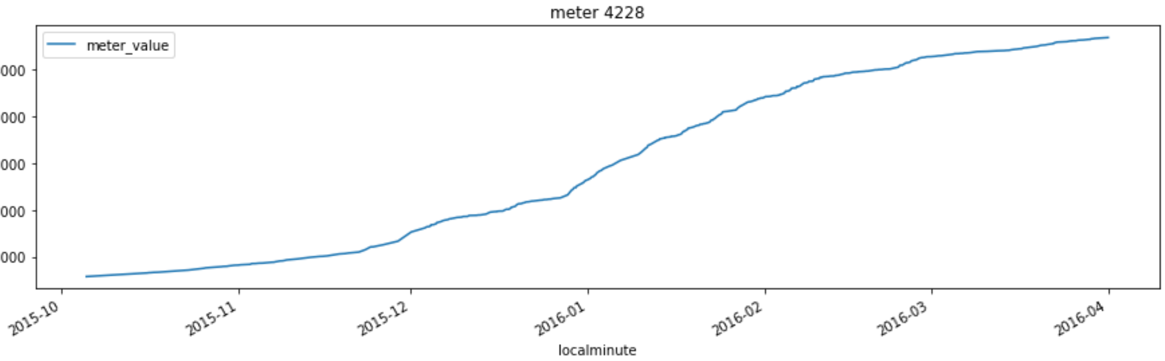

meter 3893
meter_value
105000
102500
100000
97500
95000
92500
90000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

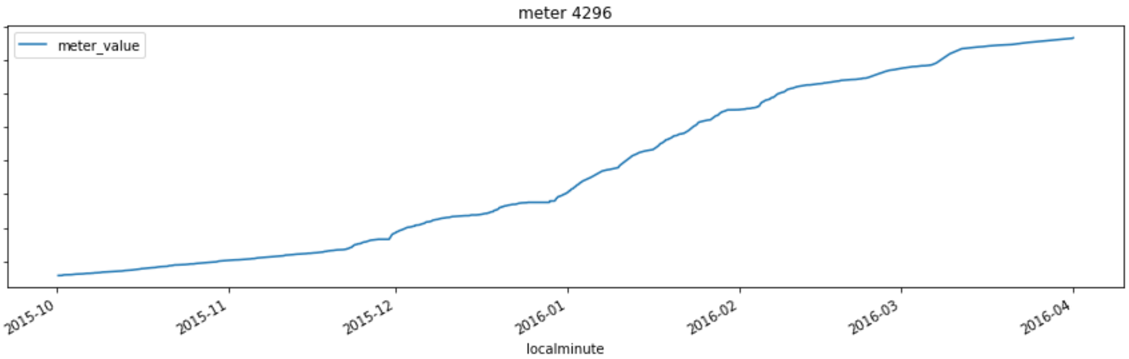

meter 3918
meter_value
290000
285000
280000
275000
270000
265000
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

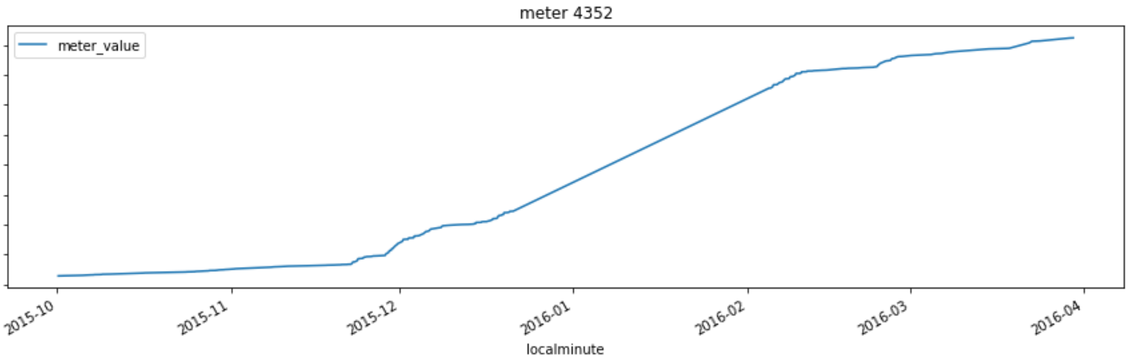

meter 4029
meter_value
320000
317500
315000
312500
310000
307500
305000
302500
300000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

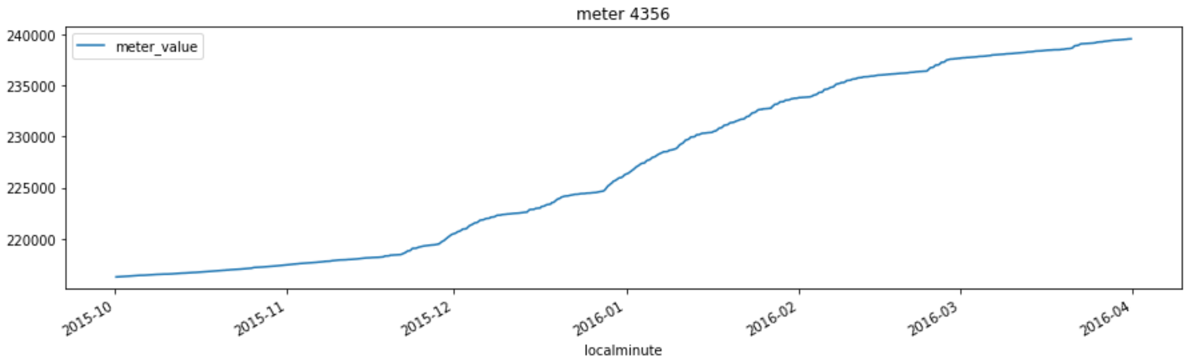

meter 4031
meter_value
120000
117500
115000
112500
110000
107500
105000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

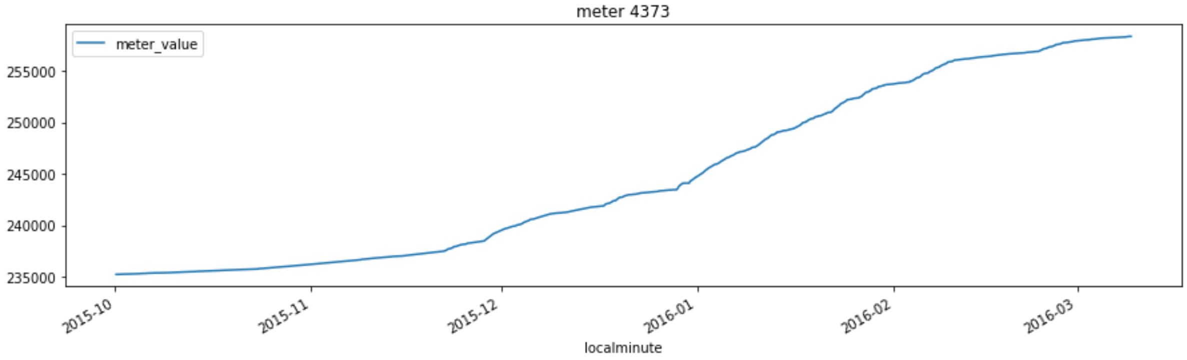

meter 4193
meter_value
315000
310000
305000
300000
295000
290000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

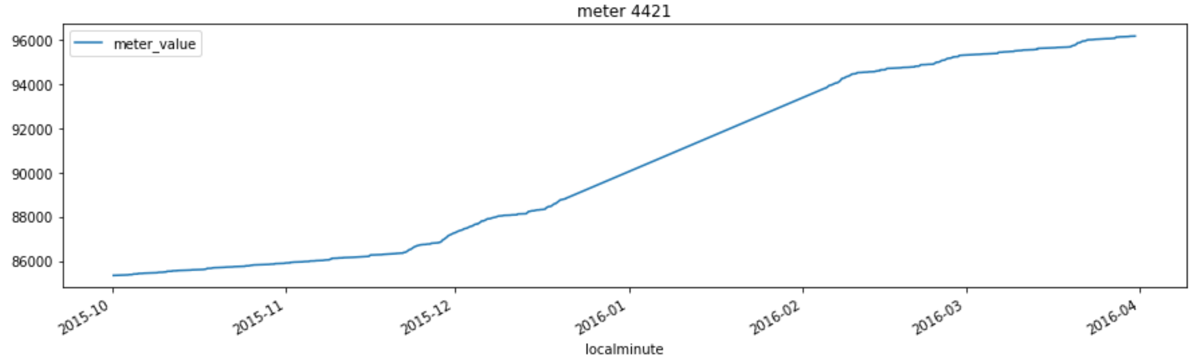

meter 4228
meter_value
94000
92000
90000
88000
86000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

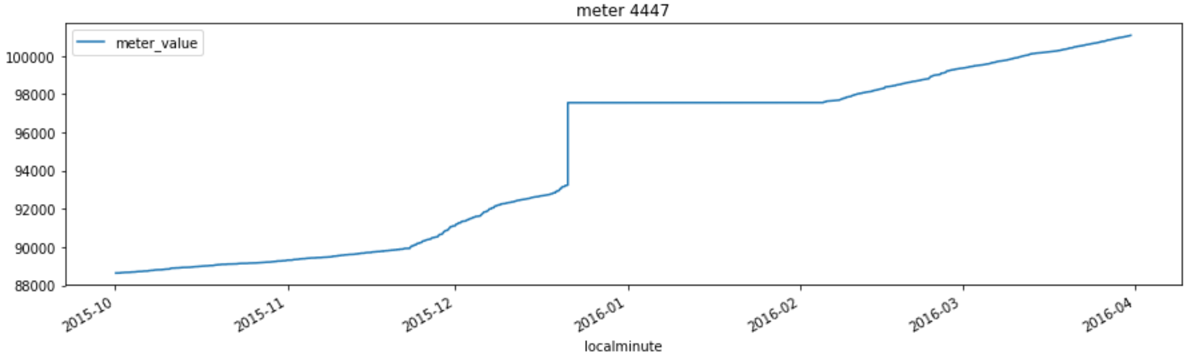

meter 4296
meter_value
187500
185000
182500
180000
177500
175000
172500
170000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

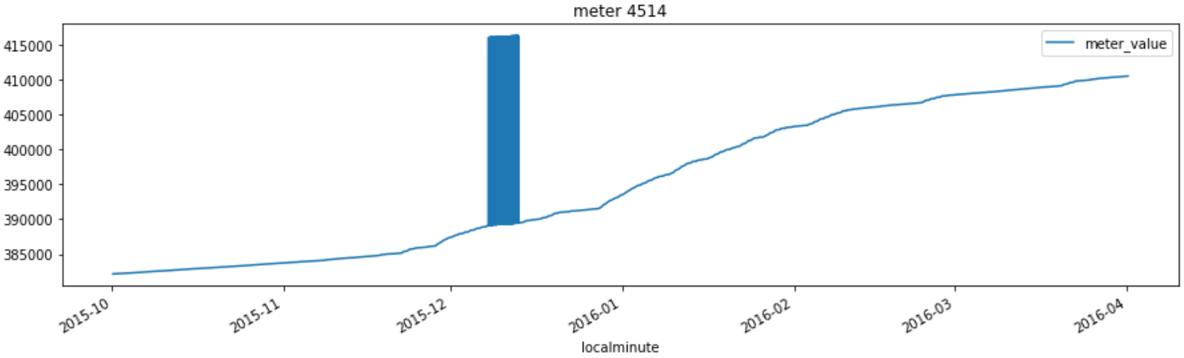

meter 4352
meter_value
237500
235000
232500
230000
227500
225000
222500
220000
217500
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

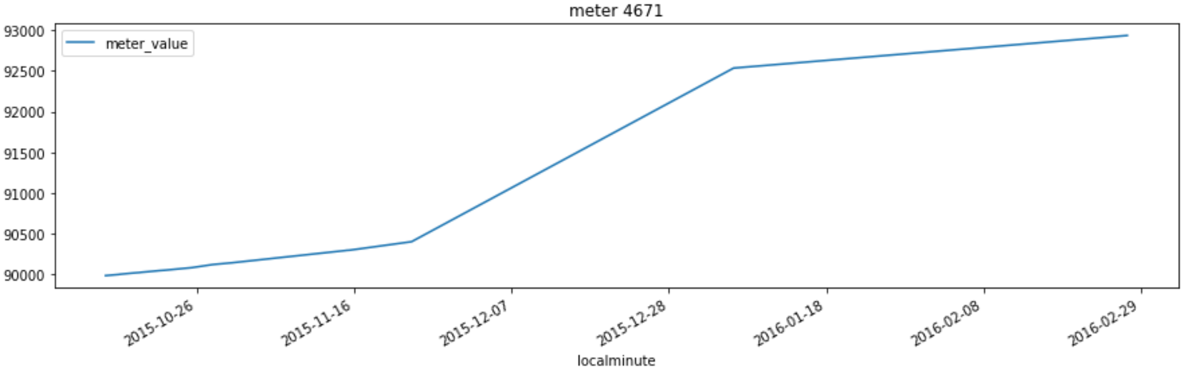

meter 4356
meter_value
240000
235000
230000
225000
220000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

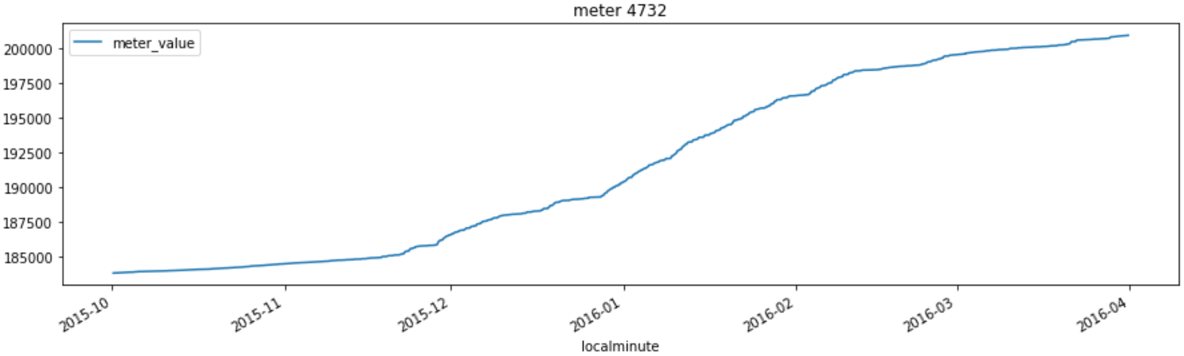

meter 4373
meter_value
255000
250000
245000
240000
235000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
localminute

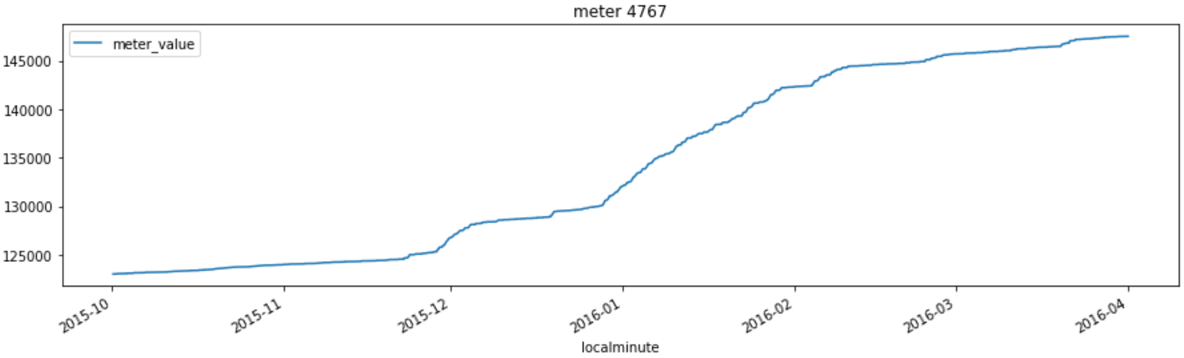

meter 4421
meter_value
96000
94000
92000
90000
88000
86000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

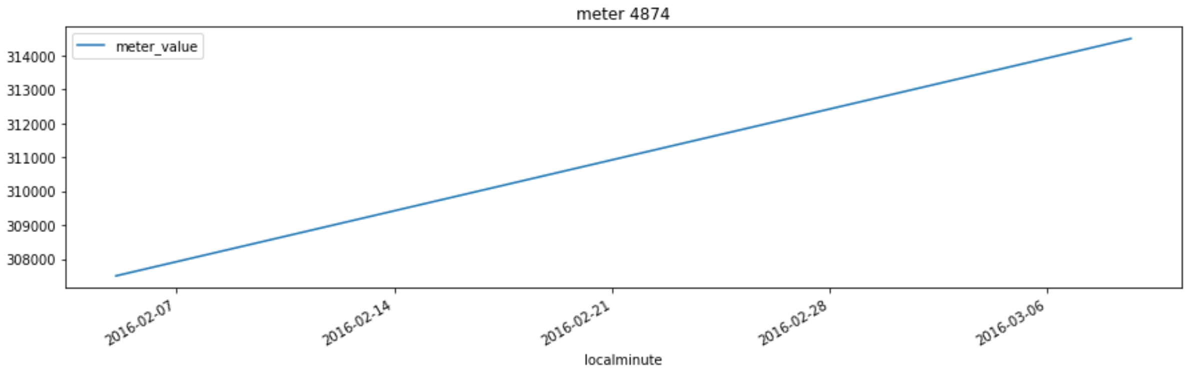

meter 4447
meter_value
100000
98000
96000
94000
92000
90000
88000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

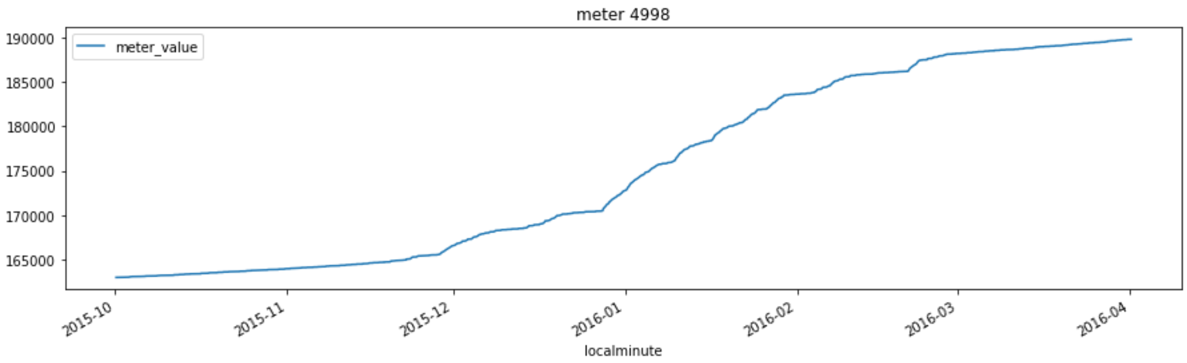

meter 4514
meter_value
415000
410000
405000
400000
395000
390000
385000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

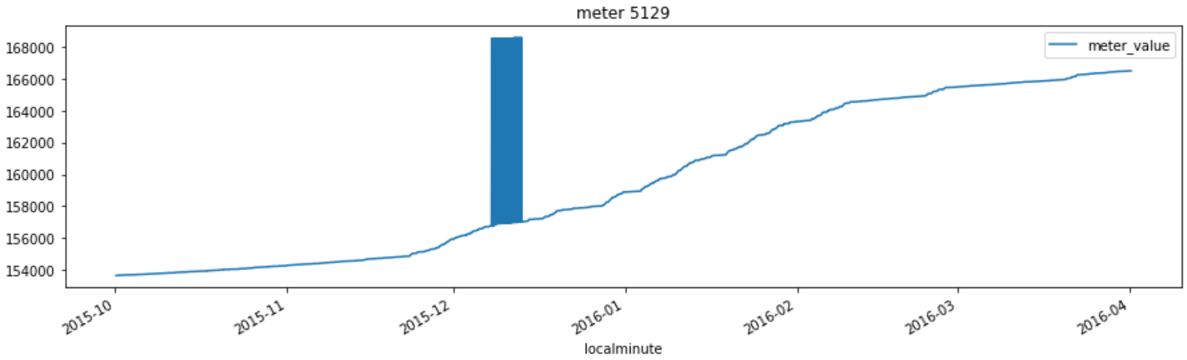

meter 4671
meter_value
93000
92500
92000
91500
91000
90500
90000
2015-10-26
2015-11-16
2015-12-07
2015-12-28
2016-01-18
2016-02-08
2016-02-29
localminute

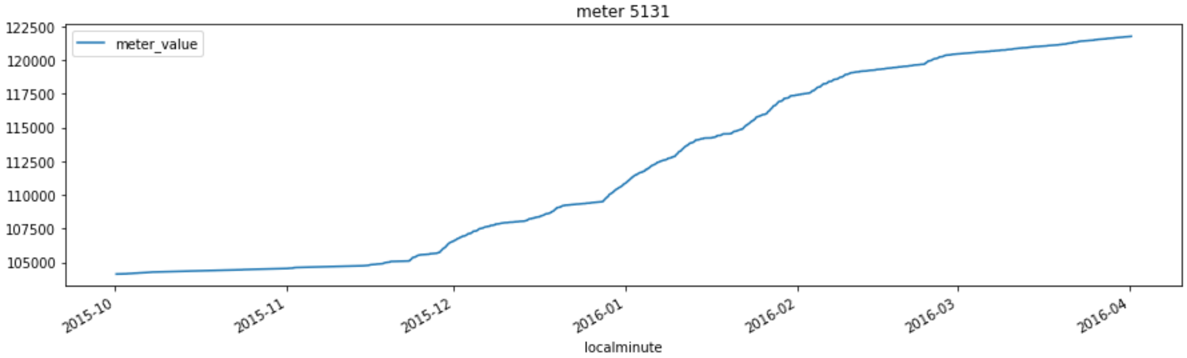

meter 4732
meter_value
200000
197500
195000
192500
190000
187500
185000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

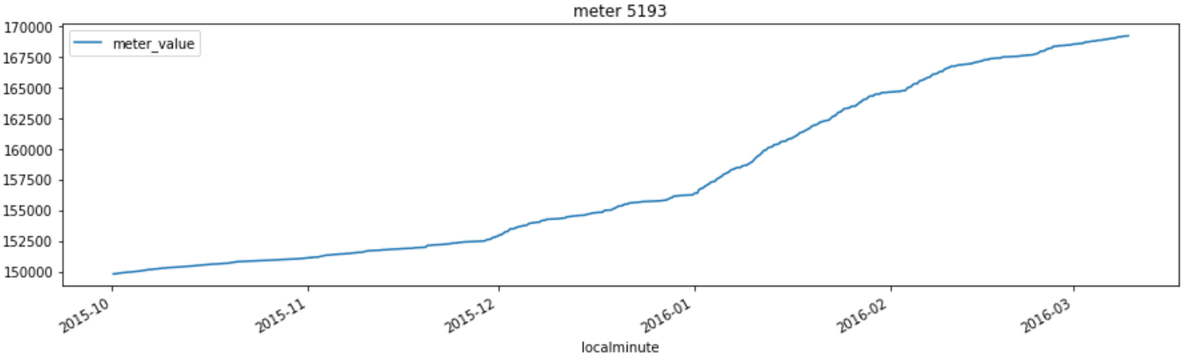

meter 4767
meter_value
145000
140000
135000
130000
125000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

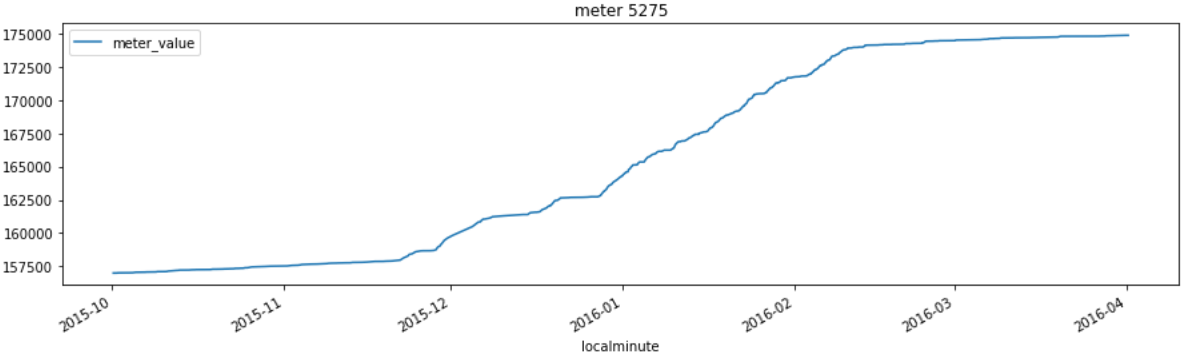

meter 4874
meter_value
314000
313000
312000
311000
310000
309000
308000
2016-02-07
2016-02-14
2016-02-21
2016-02-28
2016-03-06
localminute

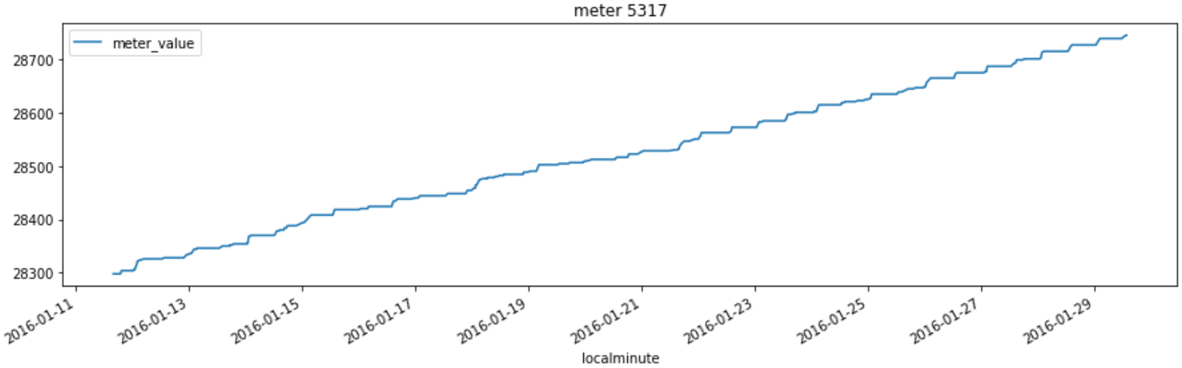

meter 4998
meter_value
190000
185000
180000
175000
170000
165000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

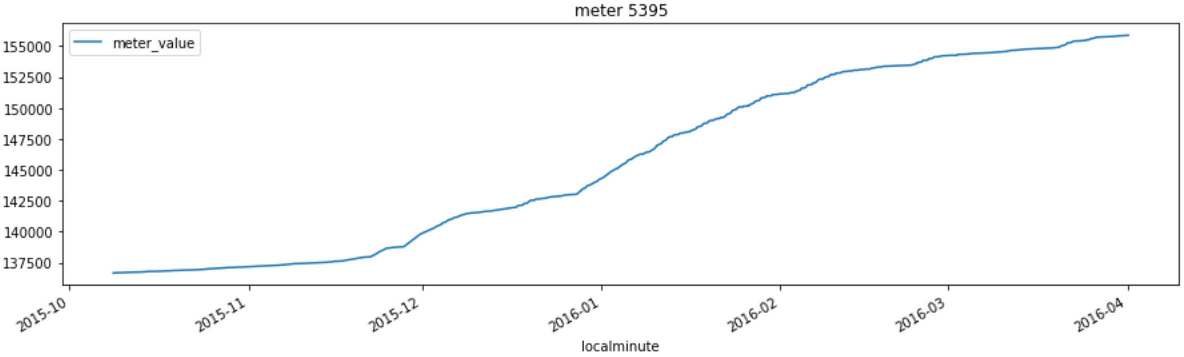

meter 5129
meter_value
168000
166000
164000
162000
160000
158000
156000
154000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

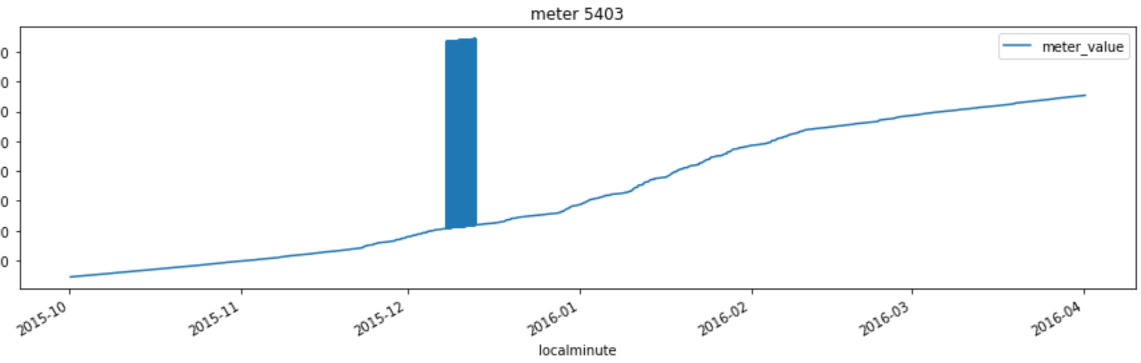

meter 5131
meter_value
122500
120000
117500
115000
112500
110000
107500
105000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

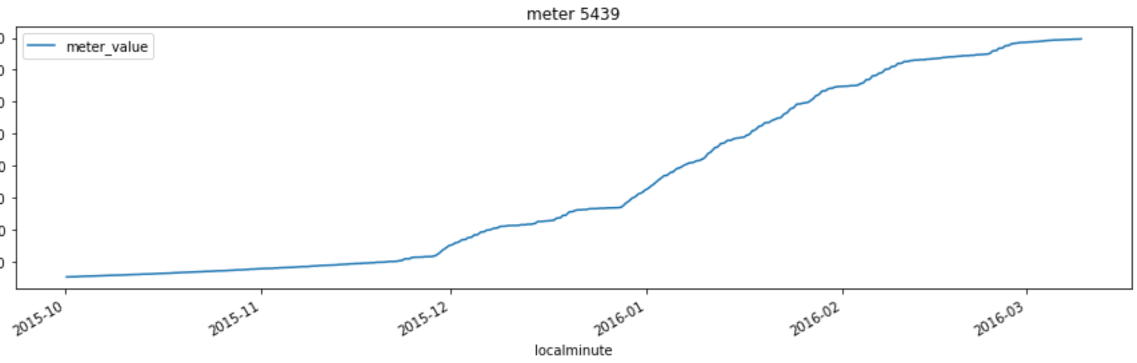

meter 5193
meter_value
170000
167500
165000
162500
160000
157500
155000
152500
150000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
localminute

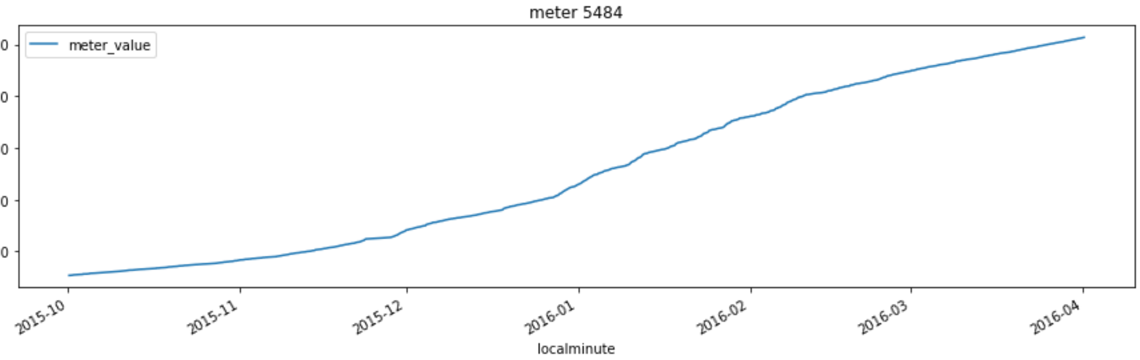

meter 5275
meter_value
175000
172500
170000
167500
165000
162500
160000
157500
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

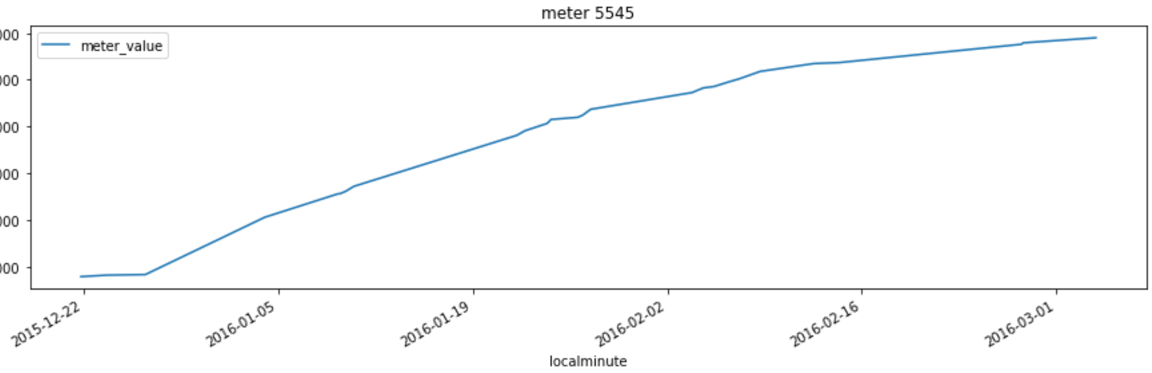

meter 5317
meter_value
28700
28600
28500
28400
28300
2016-01-11
2016-01-13
2016-01-15
2016-01-17
2016-01-19
2016-01-21
2016-01-23
2016-01-25
2016-01-27
2016-01-29
localminute

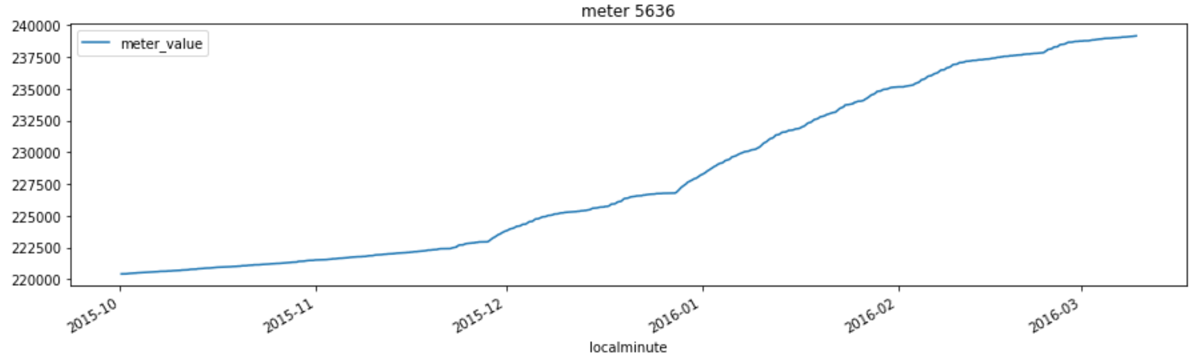

meter 5395
meter_value
155000
152500
150000
147500
145000
142500
140000
137500
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

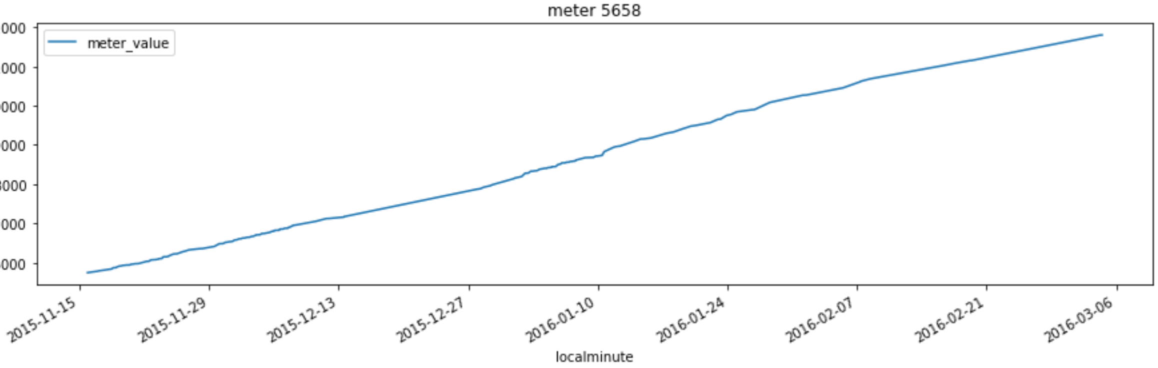

meter 5403
meter_value
128000
126000
124000
122000
120000
118000
116000
114000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

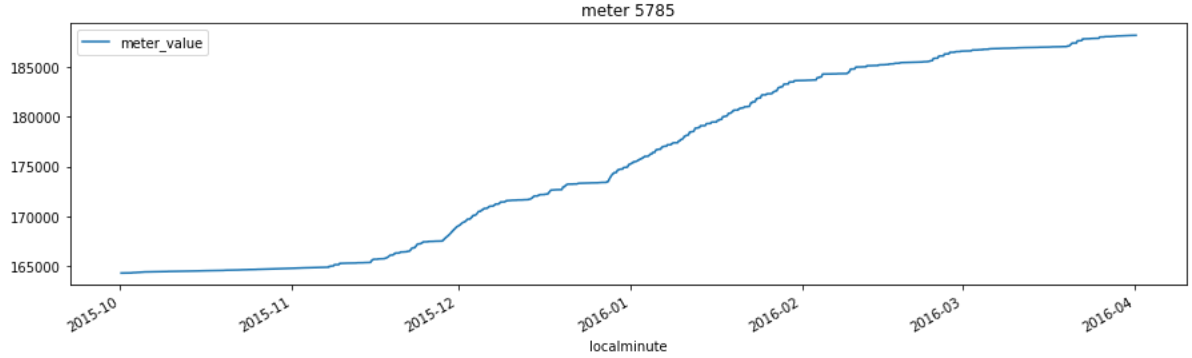

meter 5439
meter_value
196000
194000
192000
190000
188000
186000
184000
182000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
localminute

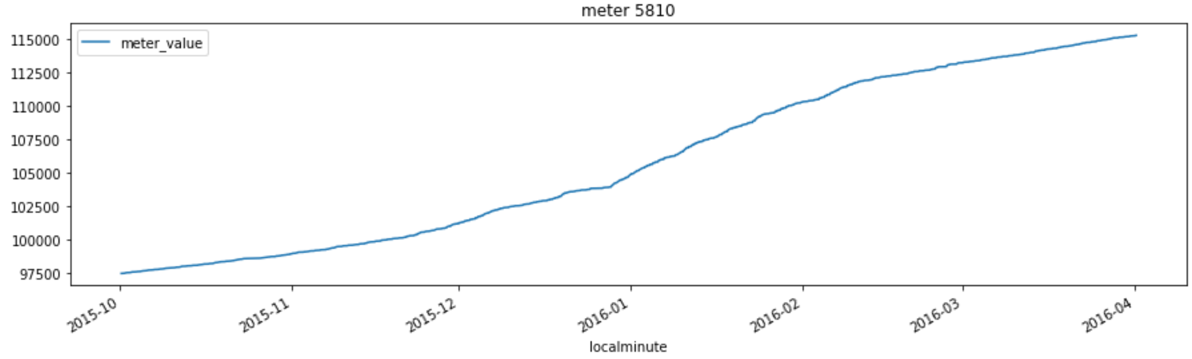

meter 5484
meter_value
285000
280000
275000
270000
265000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

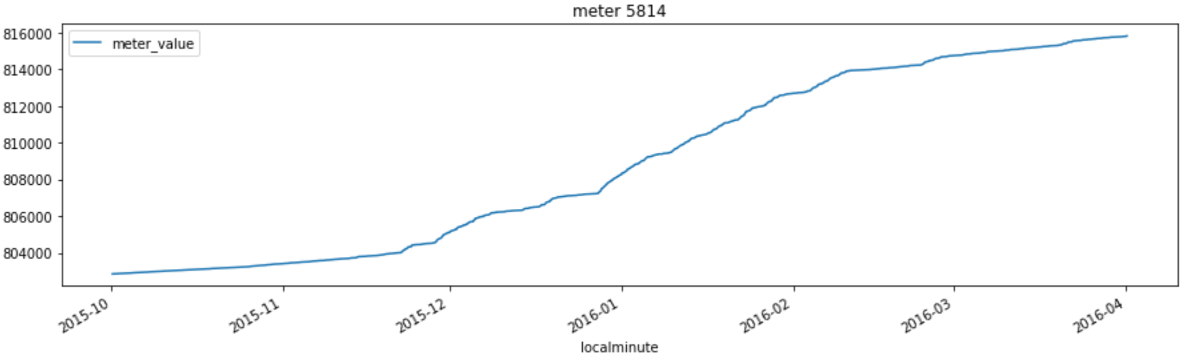

meter 5545
meter_value
150000
148000
146000
144000
142000
140000
2015-12-22
2016-01-05
2016-01-19
2016-02-02
2016-02-16
2016-03-01
localminute

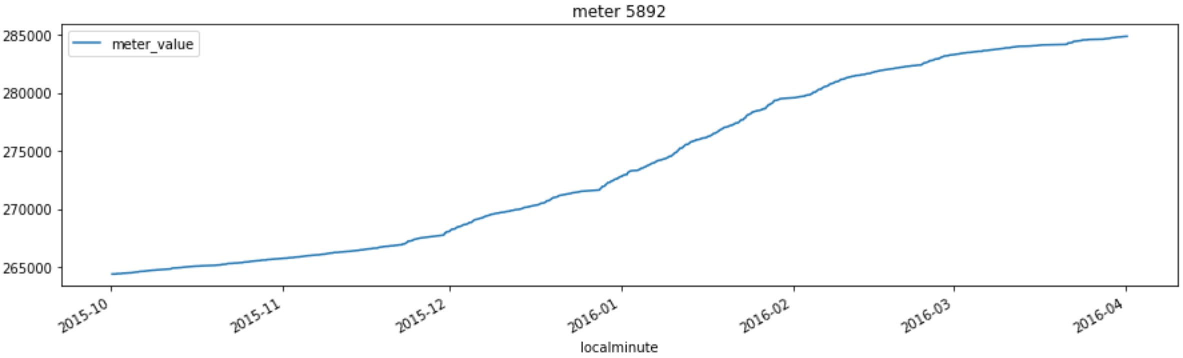

meter 5636
meter_value
240000
237500
235000
232500
230000
227500
225000
222500
220000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
localminute

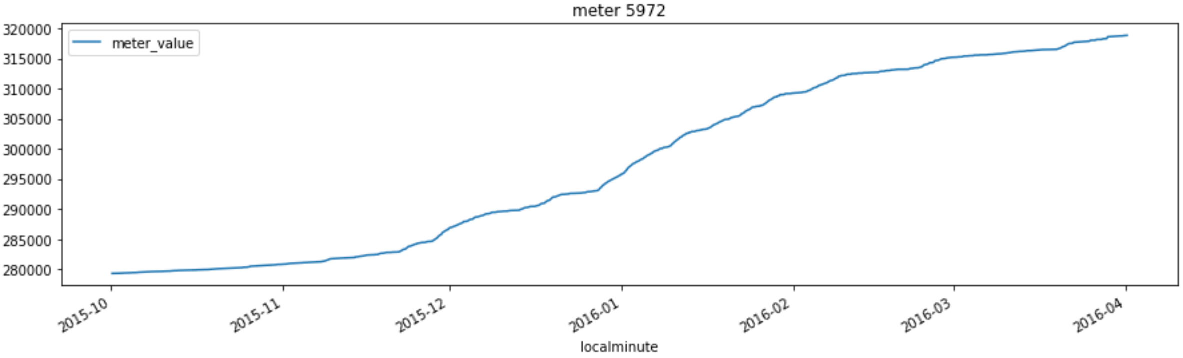

meter 5658
meter_value
152000
151000
150000
149000
148000
147000
146000
2015-11-15
2015-11-29
2015-12-13
2015-12-27
2016-01-10
2016-01-24
2016-02-07
2016-02-21
2016-03-06
localminute

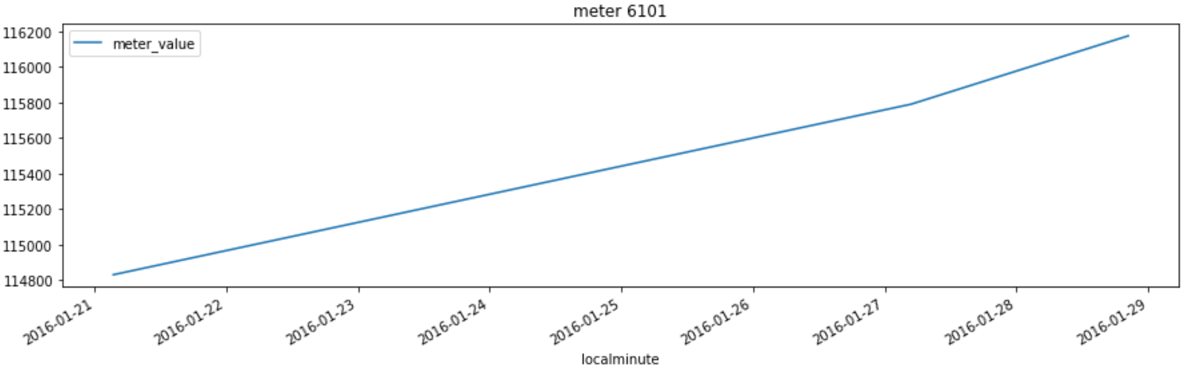

meter 5785
meter_value
185000
180000
175000
170000
165000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

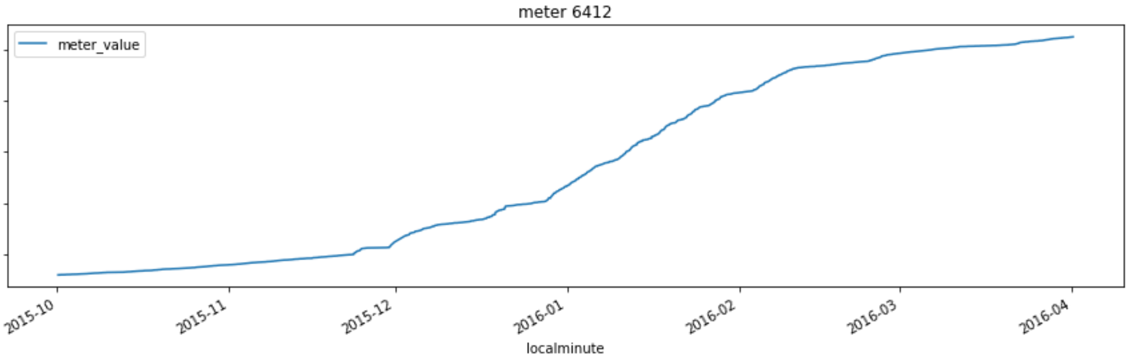

meter 5810
meter_value
115000
112500
110000
107500
105000
102500
100000
97500
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

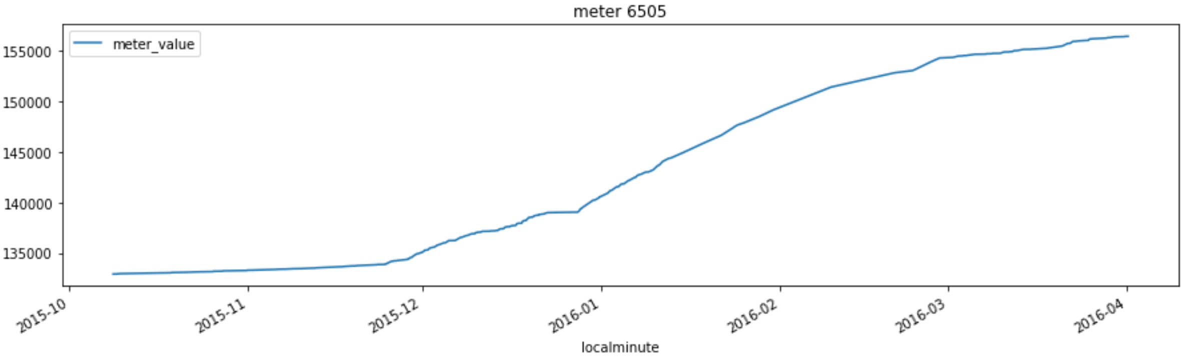

meter 5814
meter_value
816000
814000
812000
810000
808000
806000
804000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

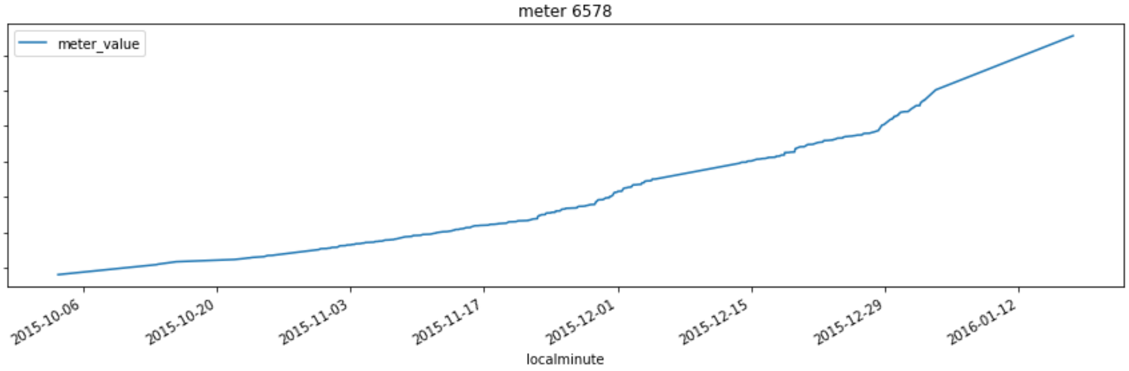

meter 5892
meter_value
285000
280000
275000
270000
265000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

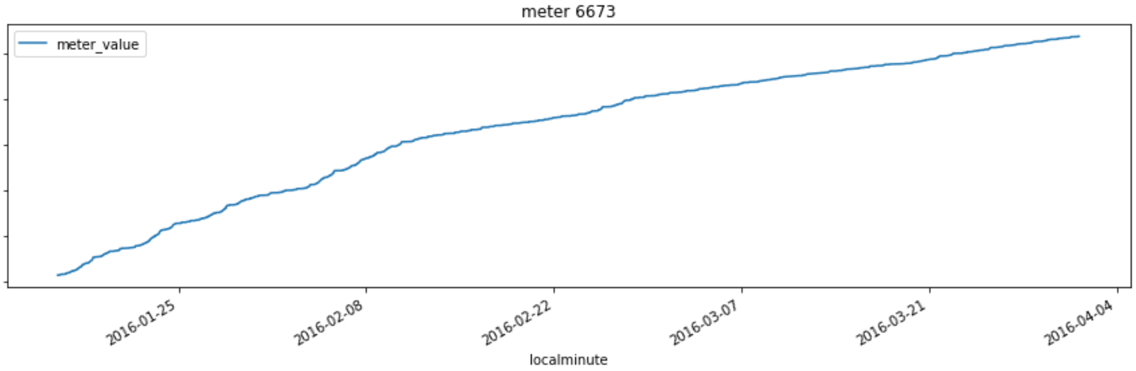

meter 5972
meter_value
320000
315000
310000
305000
300000
295000
290000
285000
280000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

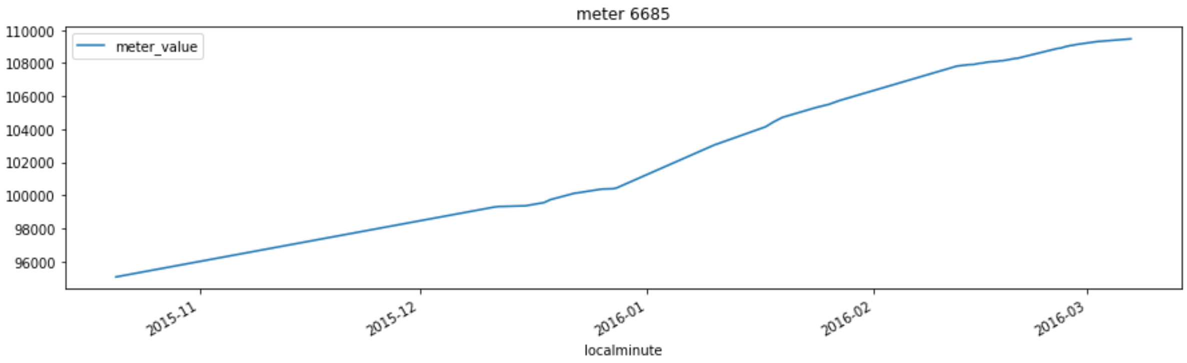

meter 6101
meter_value
116200
116000
115800
115600
115400
115200
115000
114800
2016-01-21
2016-01-22
2016-01-23
2016-01-24
2016-01-25
2016-01-26
2016-01-27
2016-01-28
2016-01-29
localminute

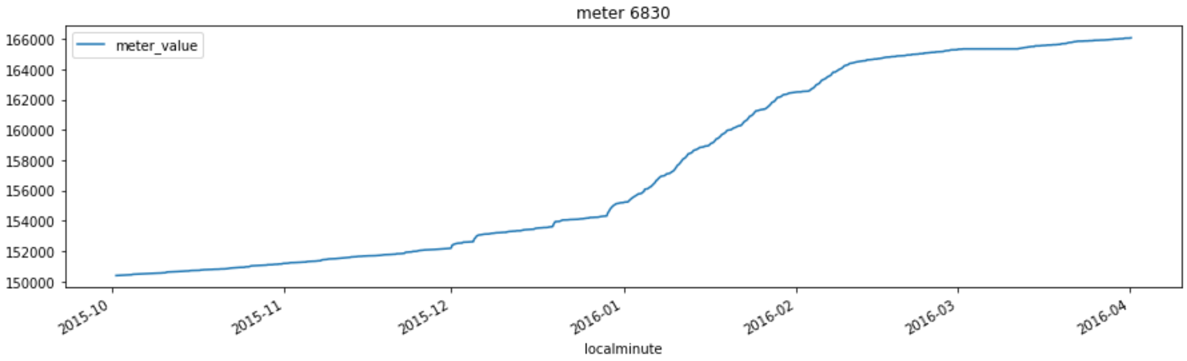

meter 6412
meter_value
155000
150000
145000
140000
135000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

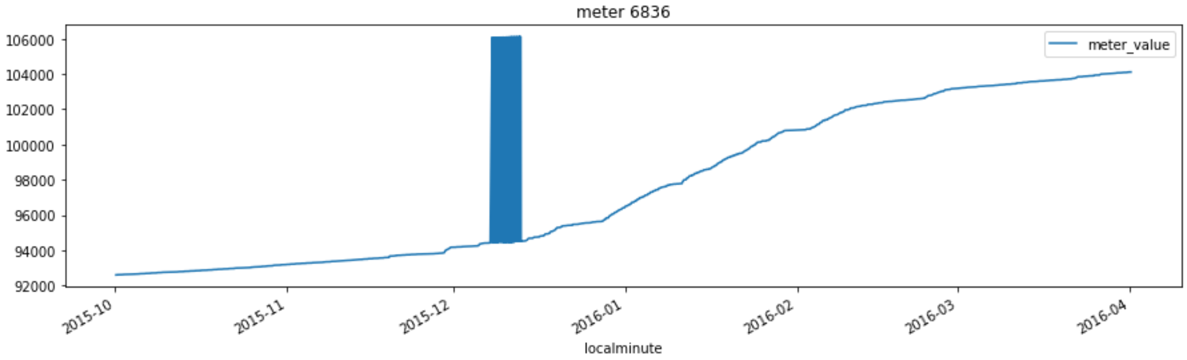

meter 6505
meter_value
155000
150000
145000
140000
135000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

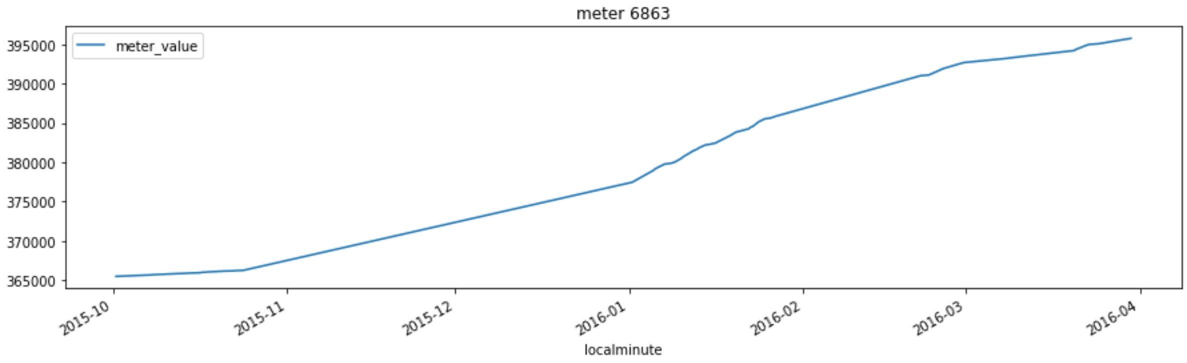

meter 6578
meter_value
164000
163000
162000
161000
160000
159000
158000
2015-10-06
2015-10-20
2015-11-03
2015-11-17
2015-12-01
2015-12-15
2015-12-29
2016-01-12
localminute

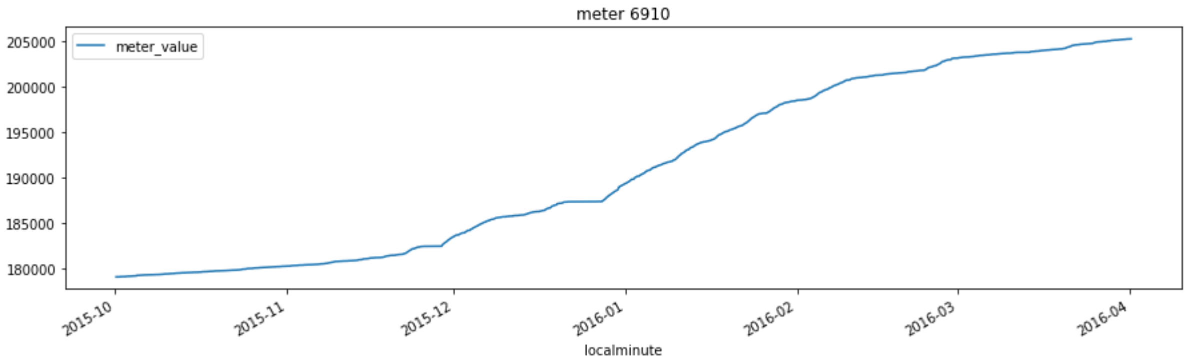

meter 6673
meter_value
85000
84000
83000
82000
81000
80000
2016-01-25
2016-02-08
2016-02-22
2016-03-07
2016-03-21
2016-04-04
localminute

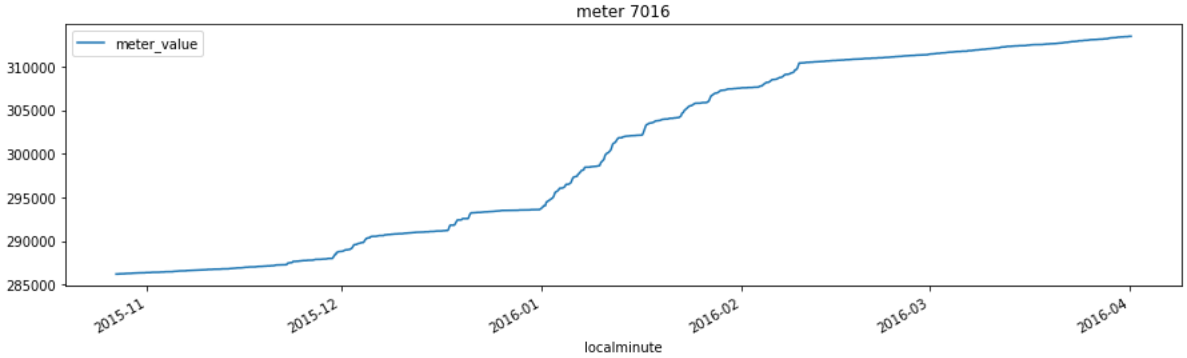

meter 6685
meter_value
110000
108000
106000
104000
102000
100000
98000
96000
2015-11
2015-12
2016-01
2016-02
2016-03
localminute

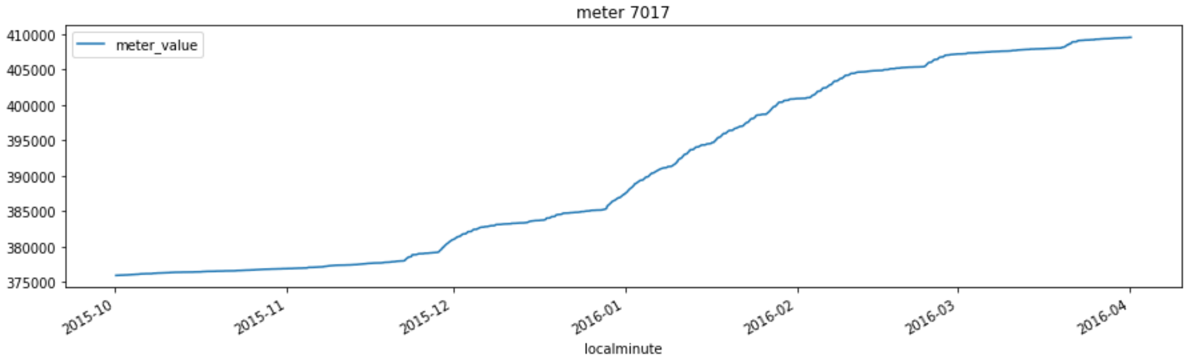

meter 6830
meter_value
166000
164000
162000
160000
158000
156000
154000
152000
150000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

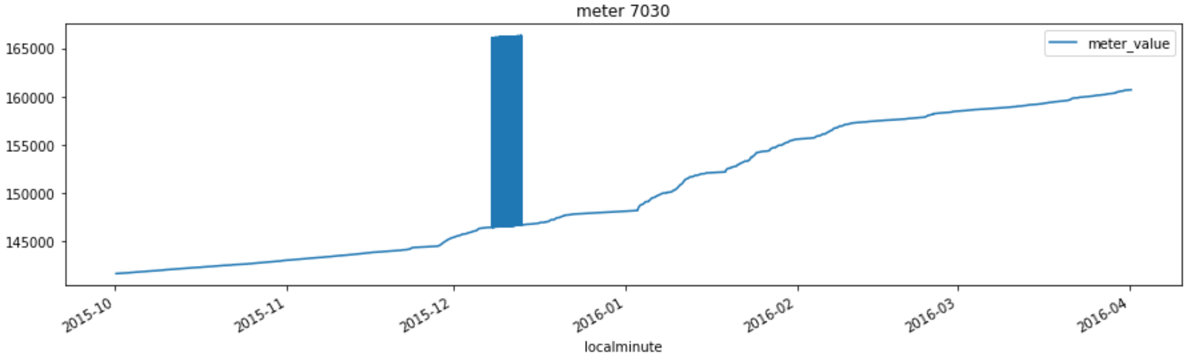

meter 6836
meter_value
106000
104000
102000
100000
98000
96000
94000
92000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

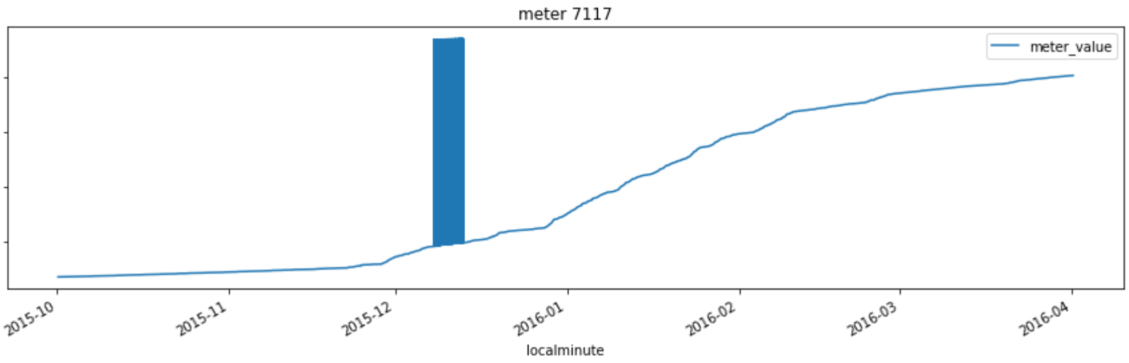

meter 6863
meter_value
395000
390000
385000
380000
375000
370000
365000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

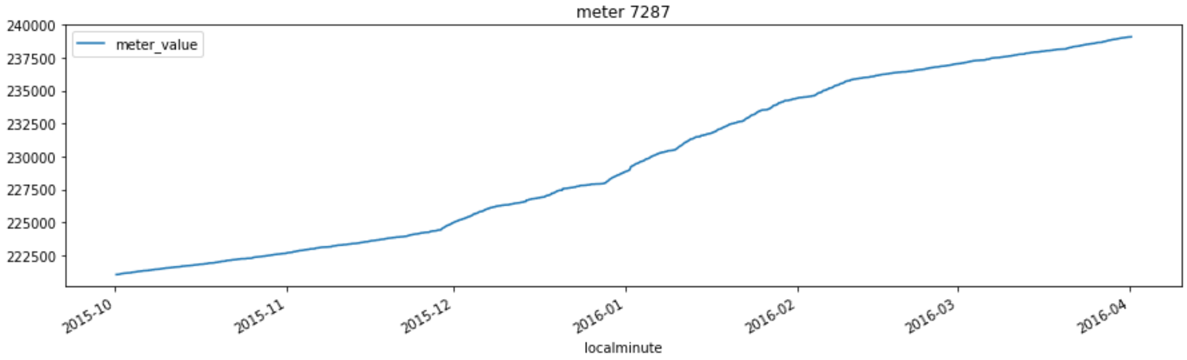

meter 6910
meter_value
205000
200000
195000
190000
185000
180000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

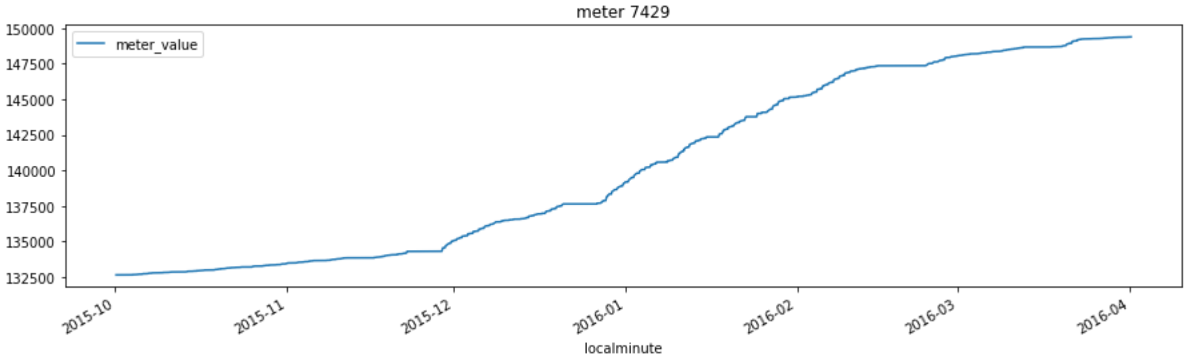

meter 7016
meter_value
310000
305000
300000
295000
290000
285000
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

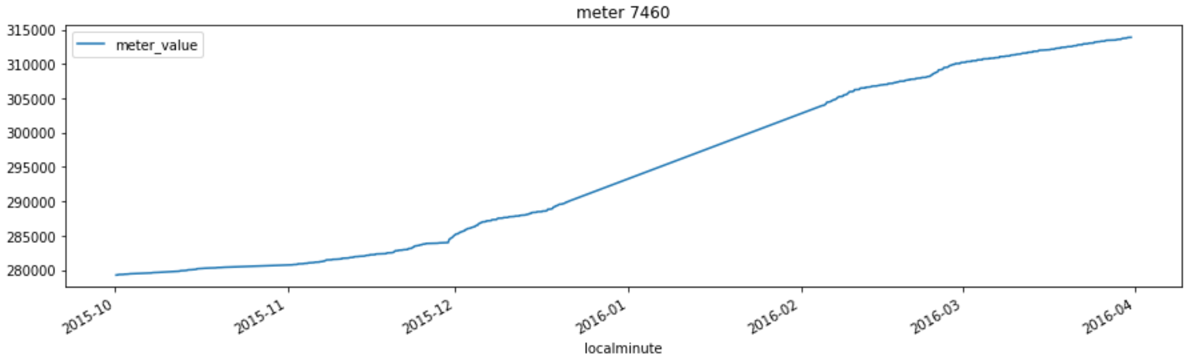

meter 7017
meter_value
410000
405000
400000
395000
390000
385000
380000
375000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

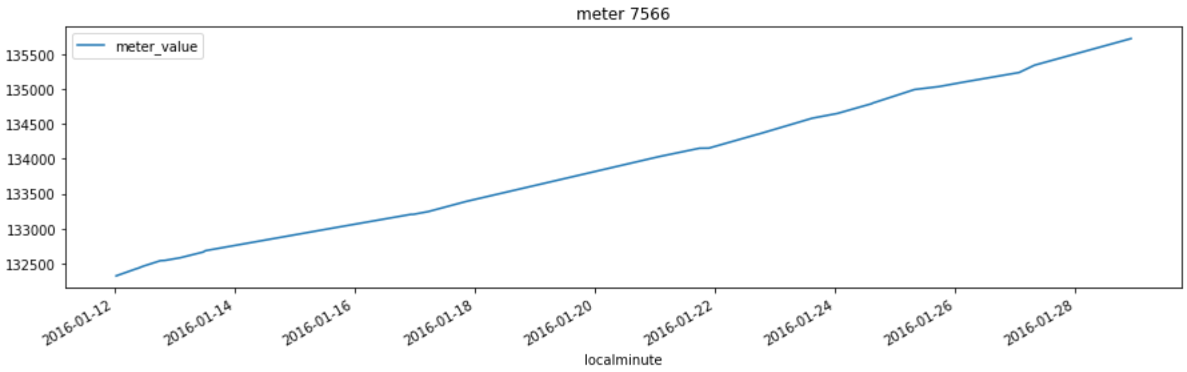

meter 7030
meter_value
165000
160000
155000
150000
145000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

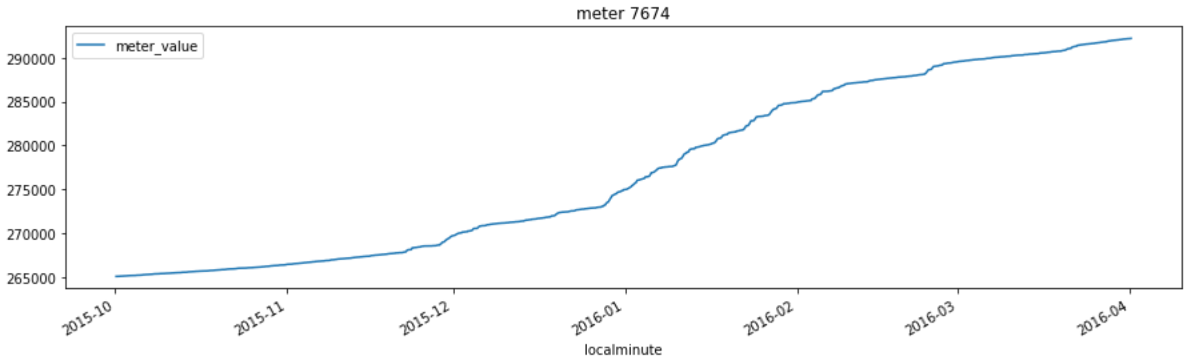

meter 7117
meter_value
230000
225000
220000
215000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

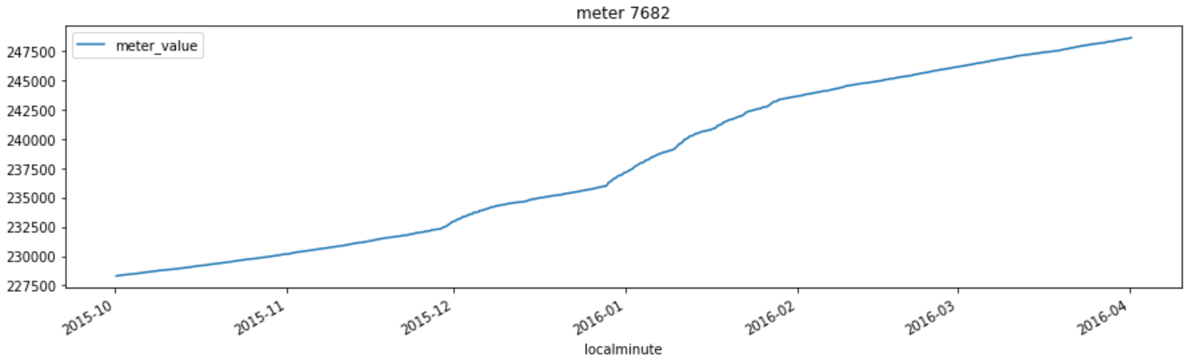

meter 7287
meter_value
240000
237500
235000
232500
230000
227500
225000
222500
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

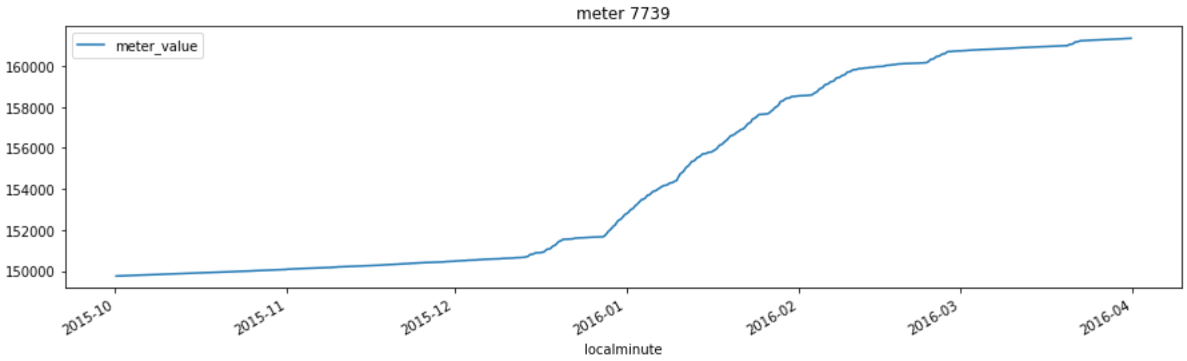

meter 7429
meter_value
150000
147500
145000
142500
140000
137500
135000
132500
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

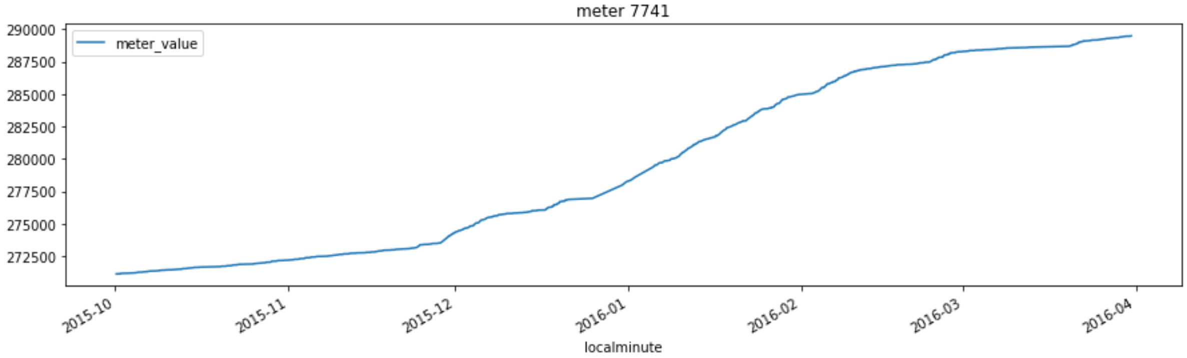

meter 7460
meter_value
315000
310000
305000
300000
295000
290000
285000
280000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

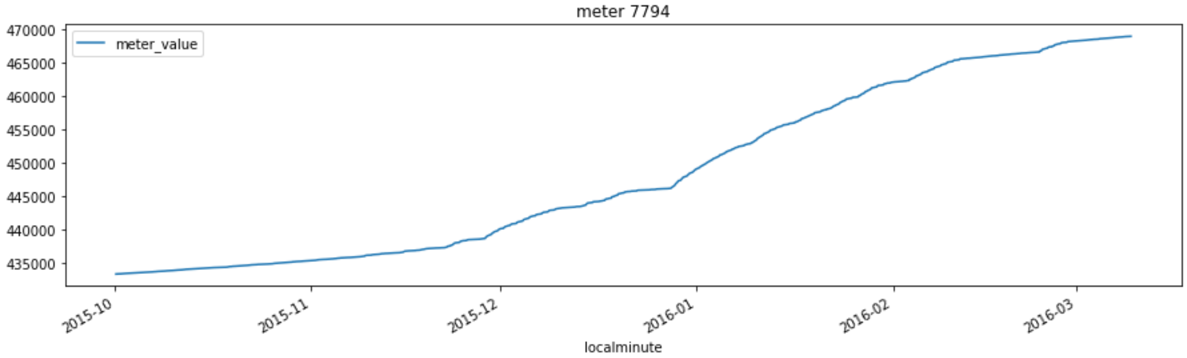

meter 7566
meter_value
135500
135000
134500
134000
133500
133000
132500
2016-01-12
2016-01-14
2016-01-16
2016-01-18
2016-01-20
2016-01-22
2016-01-24
2016-01-26
2016-01-28
localminute

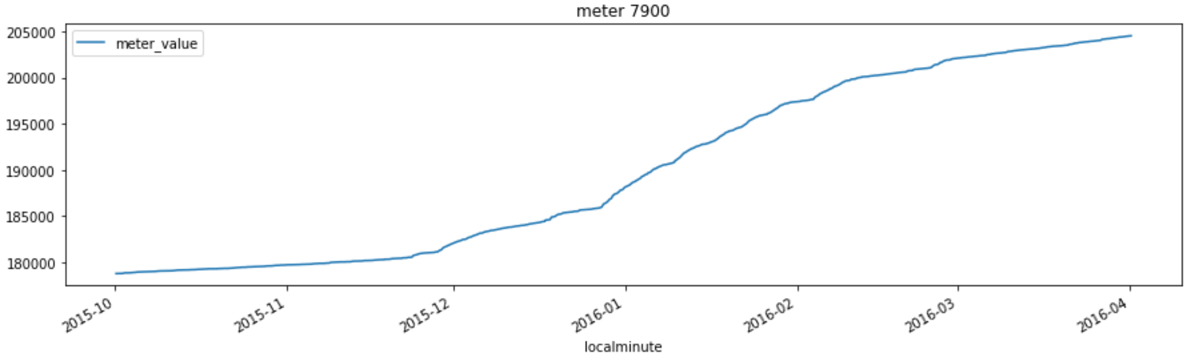

meter 7674
meter_value
290000
285000
280000
275000
270000
265000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

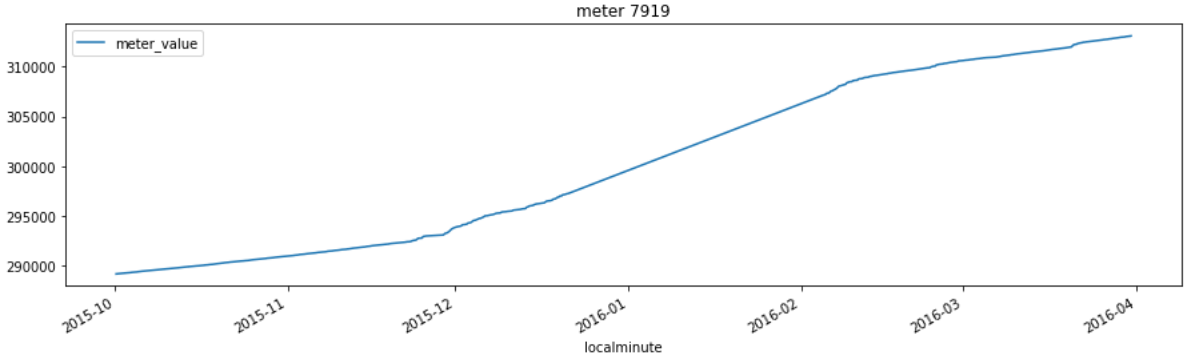

meter 7682
meter_value
247500
245000
242500
240000
237500
235000
232500
230000
227500
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

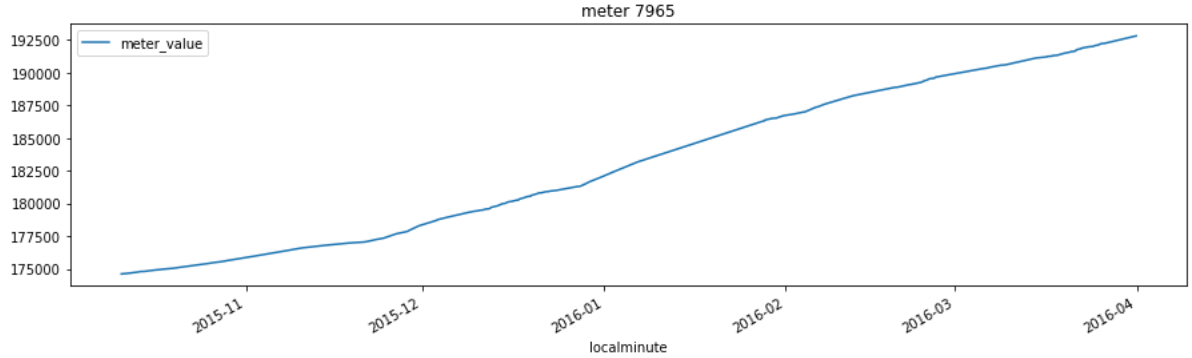

meter 7739
meter_value
160000
158000
156000
154000
152000
150000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

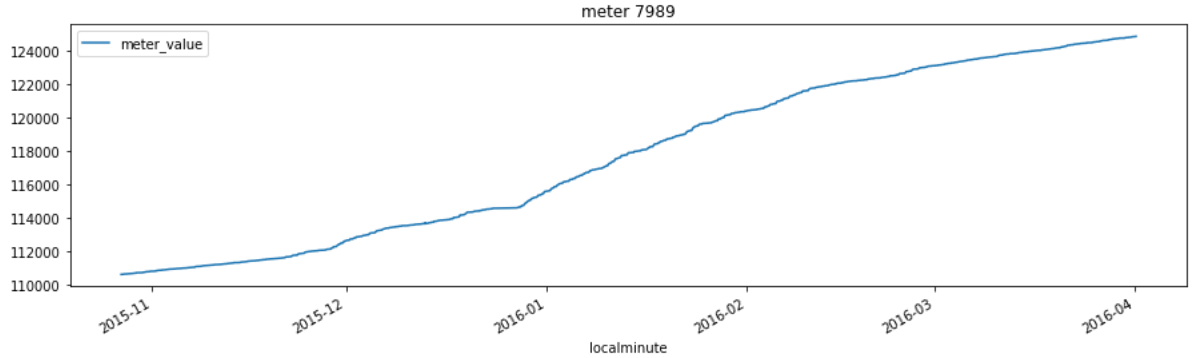

meter 7741
meter_value
290000
287500
285000
282500
280000
277500
275000
272500
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

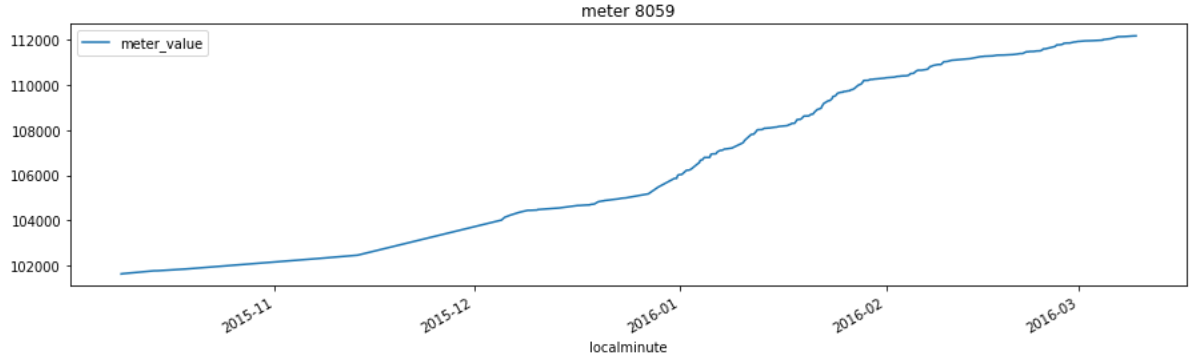

meter 7794
meter_value
470000
465000
460000
455000
450000
445000
440000
435000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
localminute

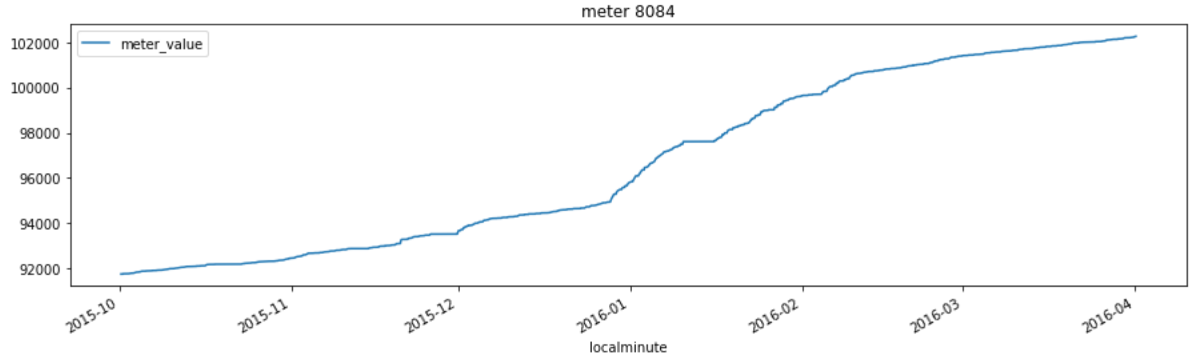

meter 7900
meter_value
205000
200000
195000
190000
185000
180000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

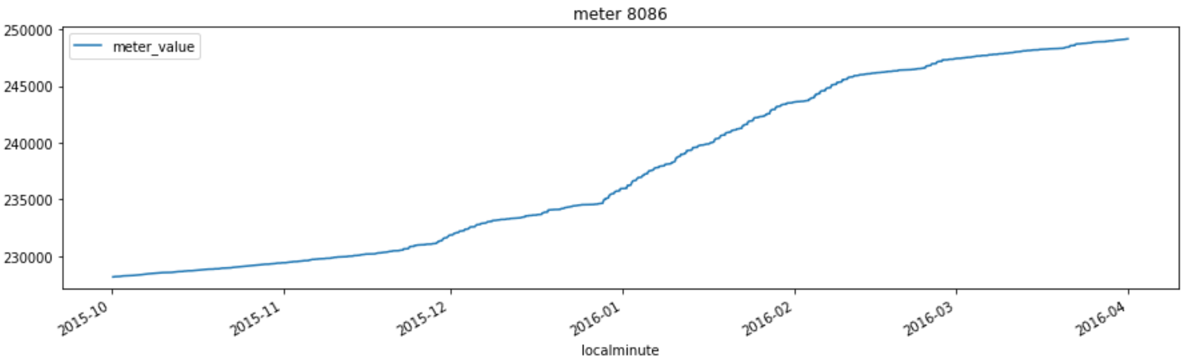

meter 7919
meter_value
310000
305000
300000
295000
290000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

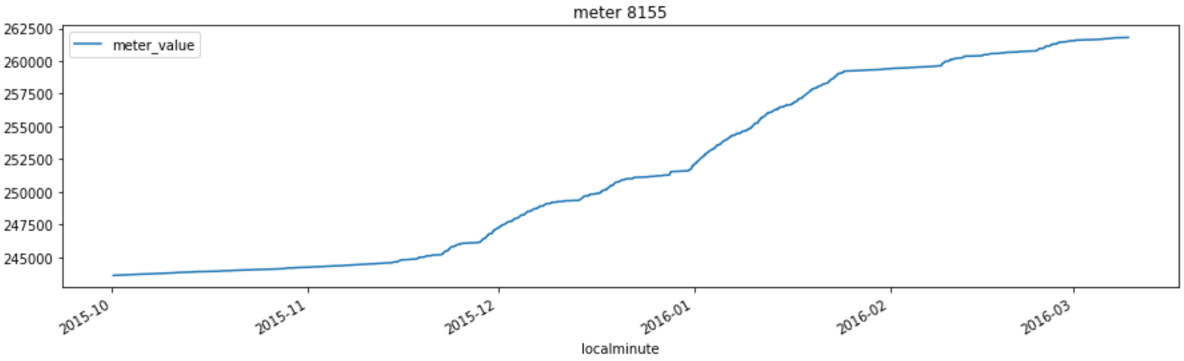

meter 7965
meter_value
192500
190000
187500
185000
182500
180000
177500
175000
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

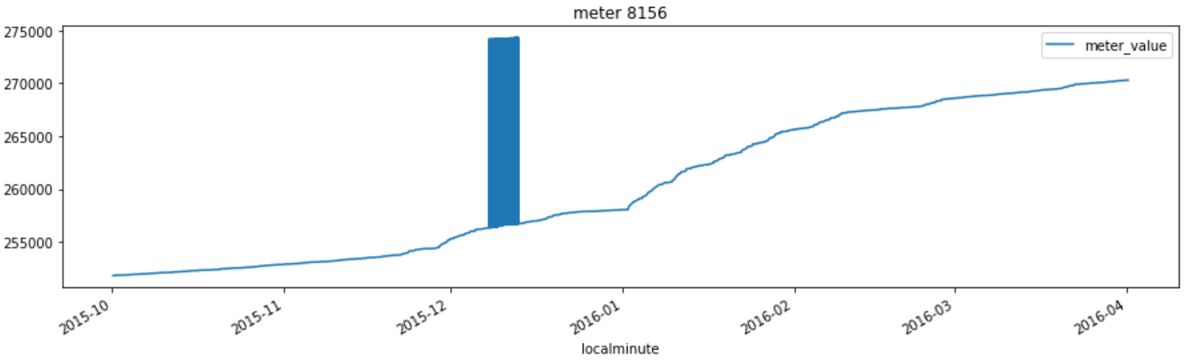

meter 7989
meter_value
124000
122000
120000
118000
116000
114000
112000
110000
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

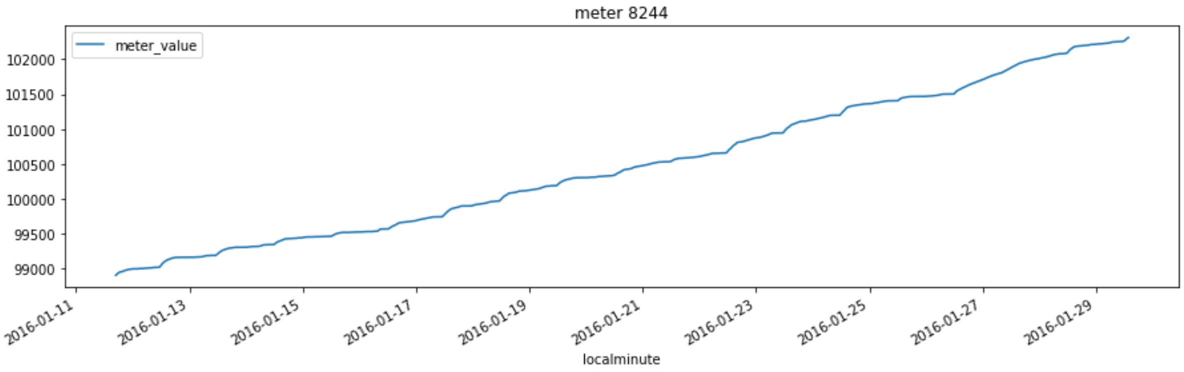

meter 8059
meter_value
112000
110000
108000
106000
104000
102000
2015-11
2015-12
2016-01
2016-02
2016-03
localminute

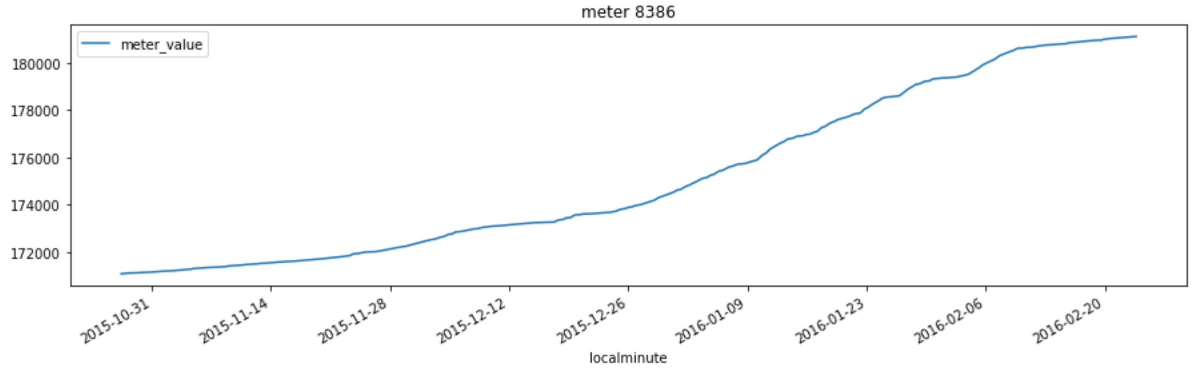

meter 8084
meter_value
102000
100000
98000
96000
94000
92000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

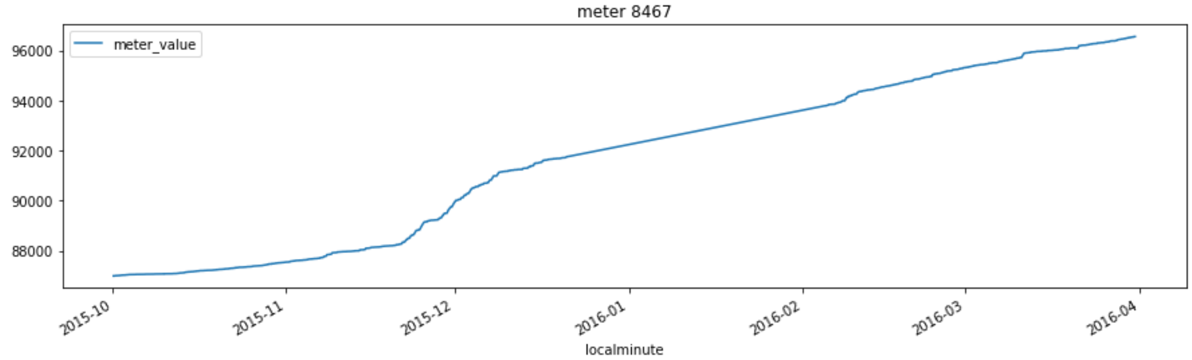

meter 8086
meter_value
250000
245000
240000
235000
230000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

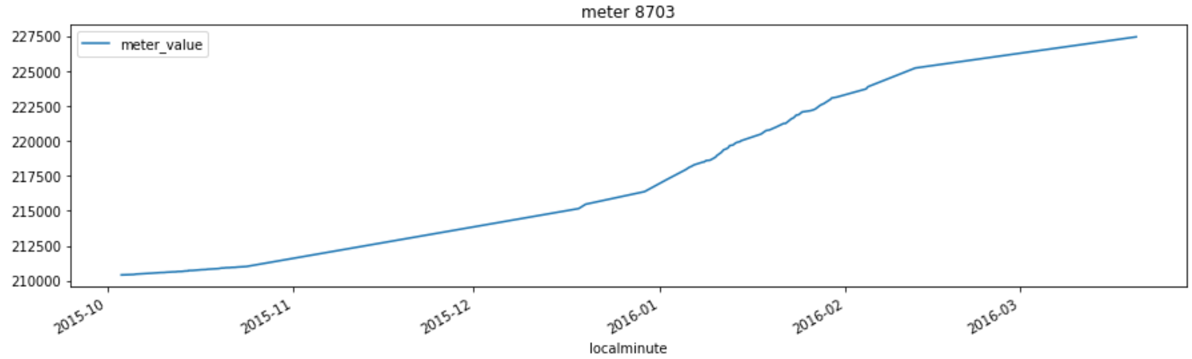

meter 8155
meter_value
262500
260000
257500
255000
252500
250000
247500
245000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
localminute

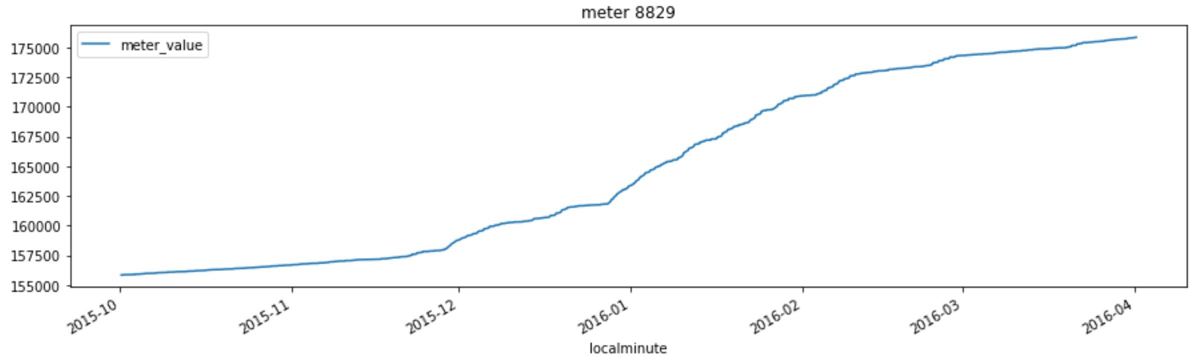

meter 8156
meter_value
275000
270000
265000
260000
255000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

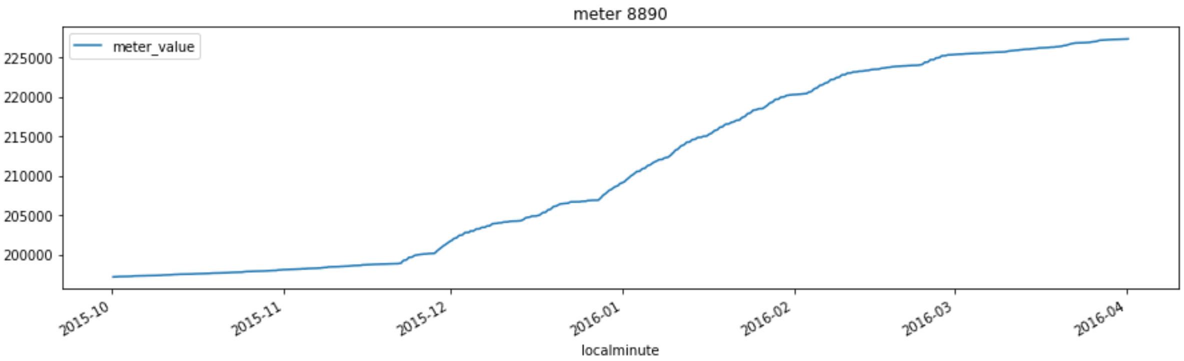

meter 8244
meter_value
102000
101500
101000
100500
100000
99500
99000
2016-01-11
2016-01-13
2016-01-15
2016-01-17
2016-01-19
2016-01-21
2016-01-23
2016-01-25
2016-01-27
2016-01-29
localminute

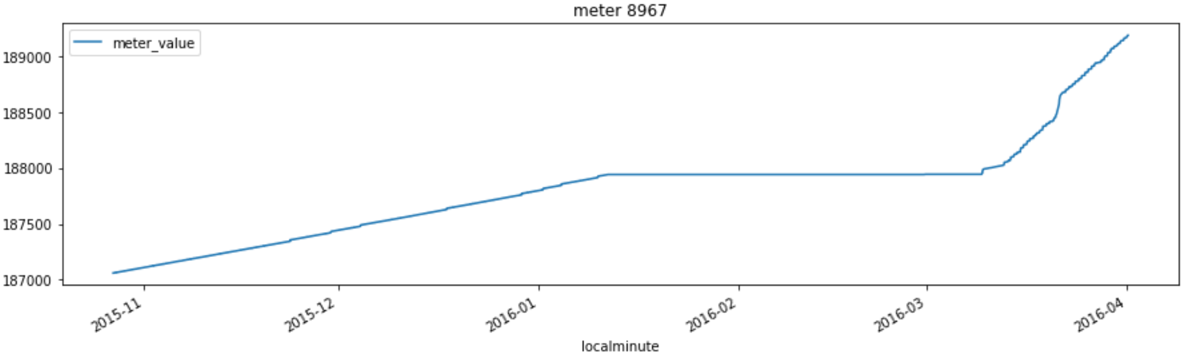

meter 8386
meter_value
180000
178000
176000
174000
172000
2015-10-31
2015-11-14
2015-11-28
2015-12-12
2015-12-26
2016-01-09
2016-01-23
2016-02-06
2016-02-20
localminute

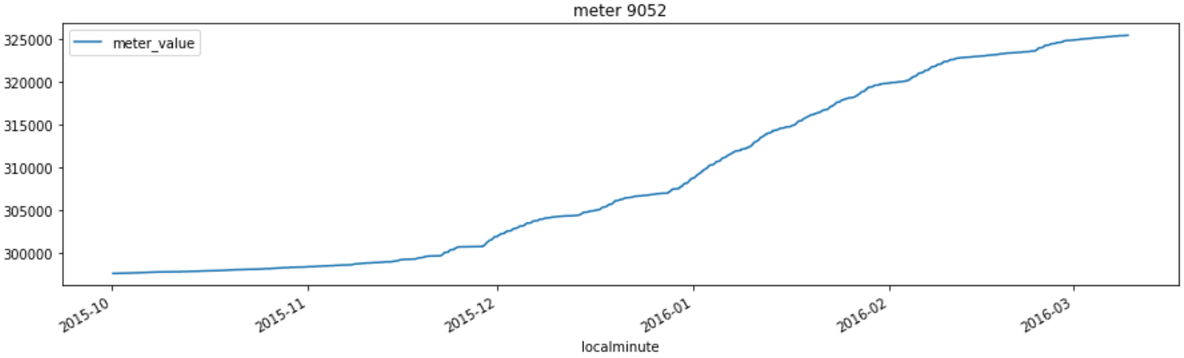

meter 8467
meter_value
96000
94000
92000
90000
88000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

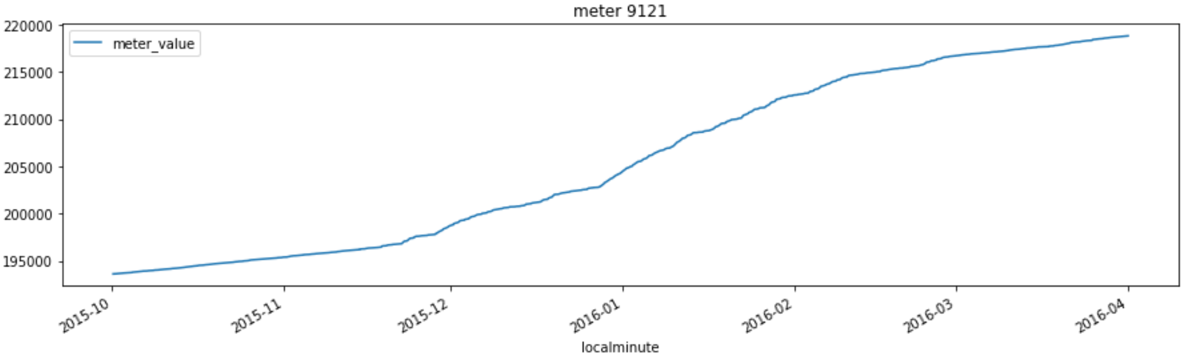

meter 8703
meter_value
227500
225000
222500
220000
217500
215000
212500
210000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
localminute

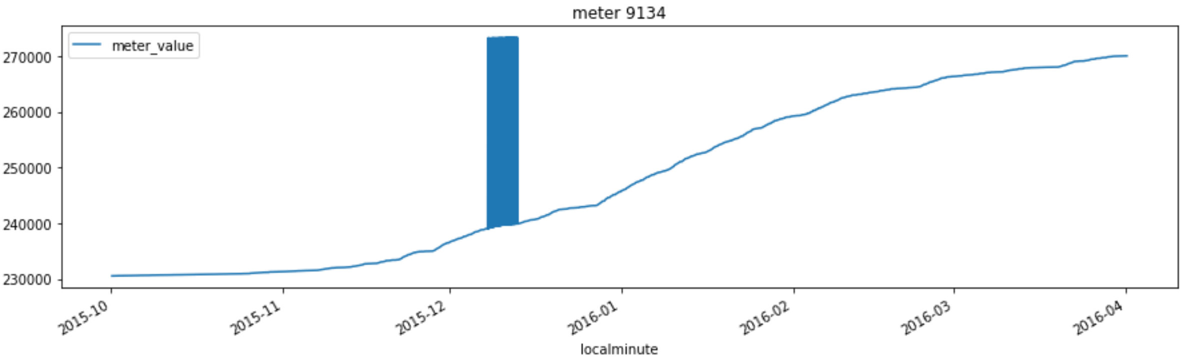

meter 8829
meter_value
175000
172500
170000
167500
165000
162500
160000
157500
155000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

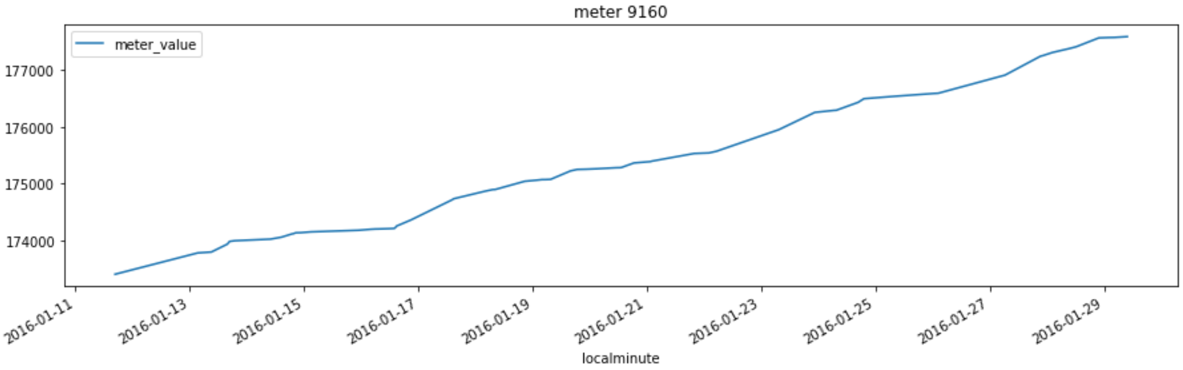

meter 8890
meter_value
225000
220000
215000
210000
205000
200000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

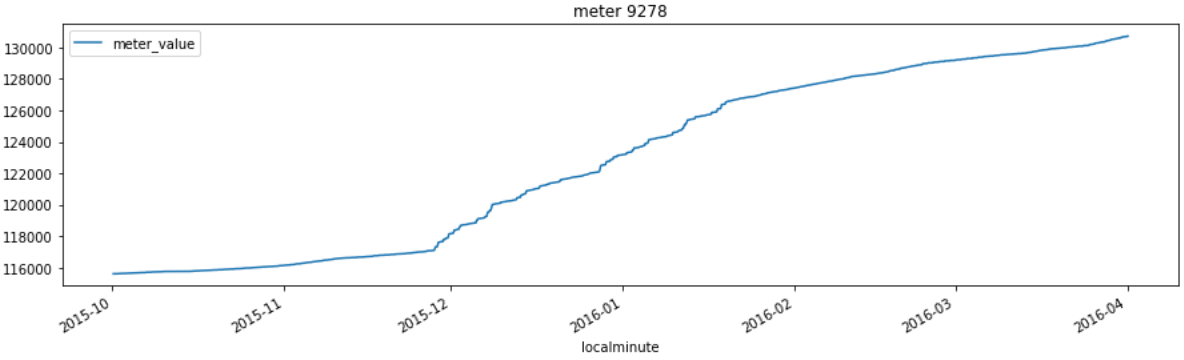

meter 8967
meter_value
189000
188500
188000
187500
187000
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

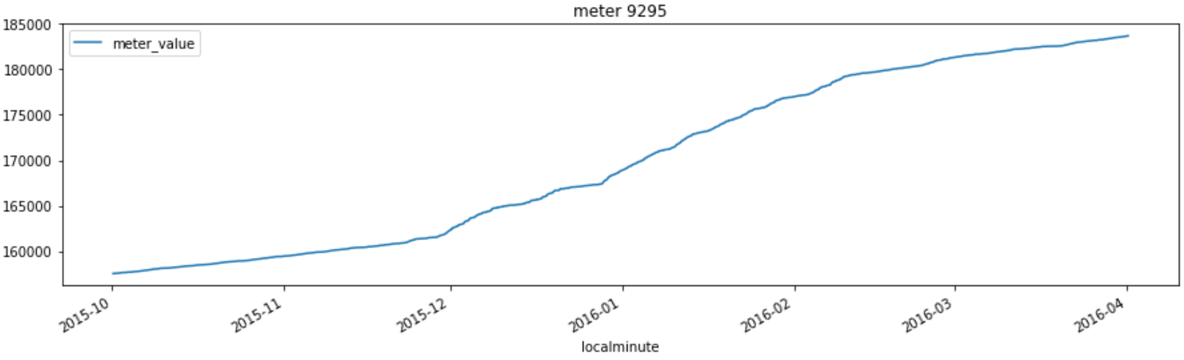

meter 9052
meter_value
325000
320000
315000
310000
305000
300000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
localminute

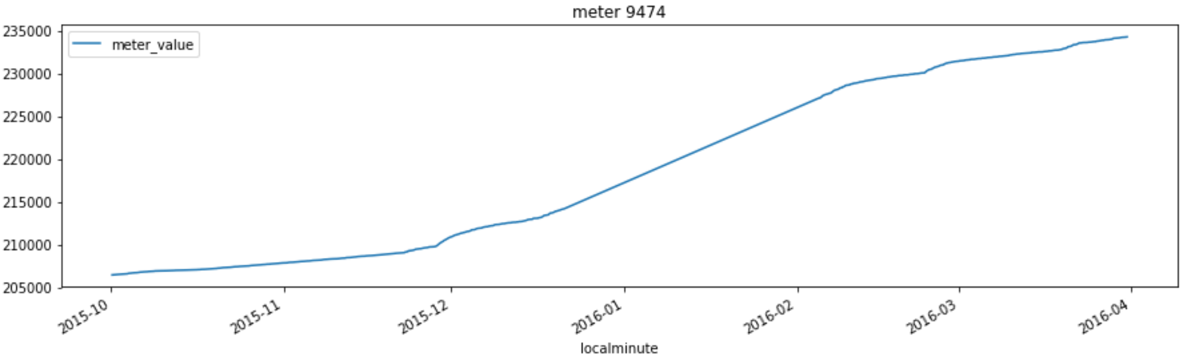

meter 9121
meter_value
220000
215000
210000
205000
200000
195000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

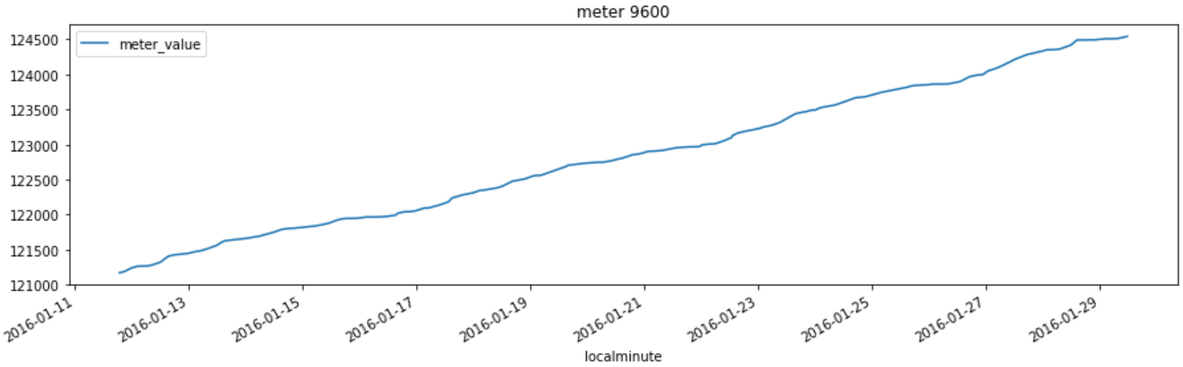

meter 9134
meter_value
270000
260000
250000
240000
230000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

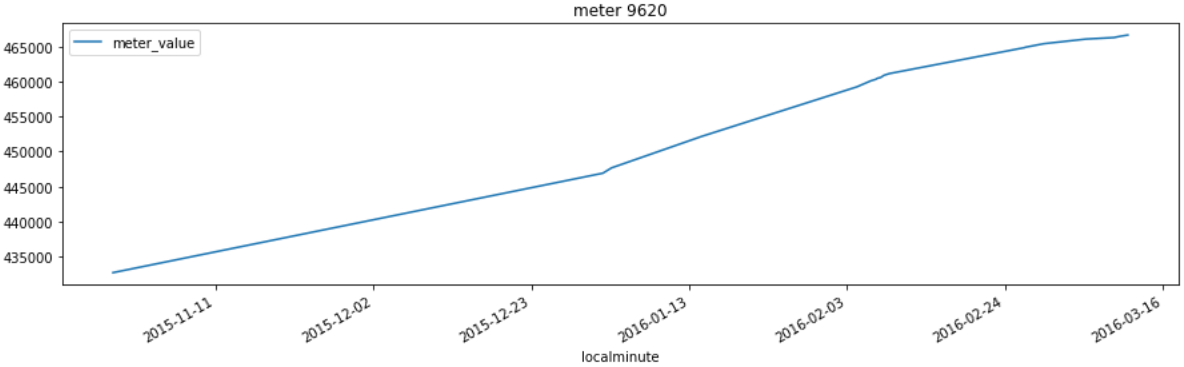

meter 9160
meter_value
177000
176000
175000
174000
2016-01-11
2016-01-13
2016-01-15
2016-01-17
2016-01-19
2016-01-21
2016-01-23
2016-01-25
2016-01-27
2016-01-29
localminute

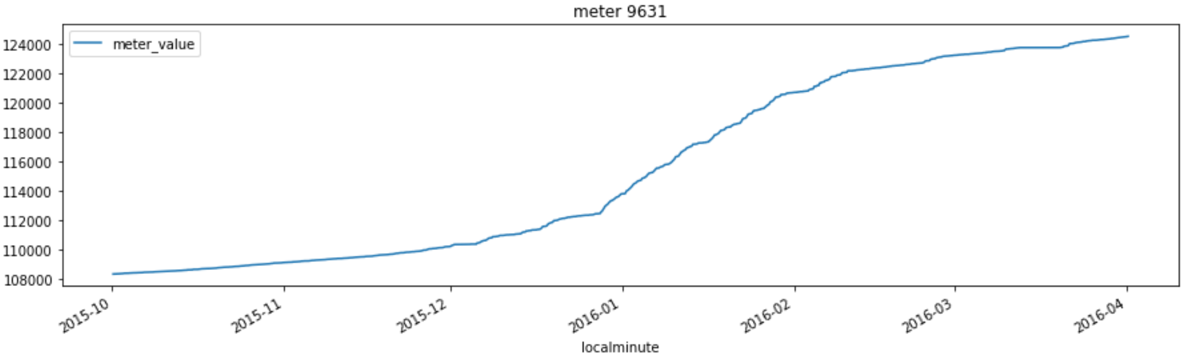

meter 9278
meter_value
130000
128000
126000
124000
122000
120000
118000
116000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

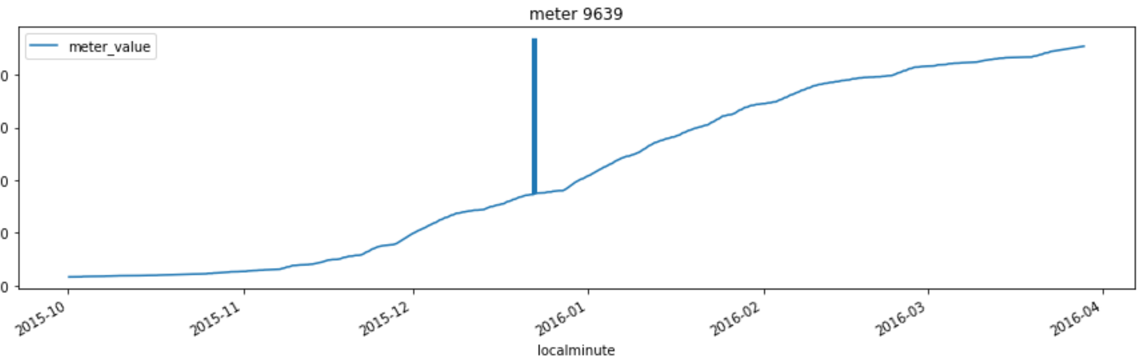

meter 9295
meter_value
185000
180000
175000
170000
165000
160000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

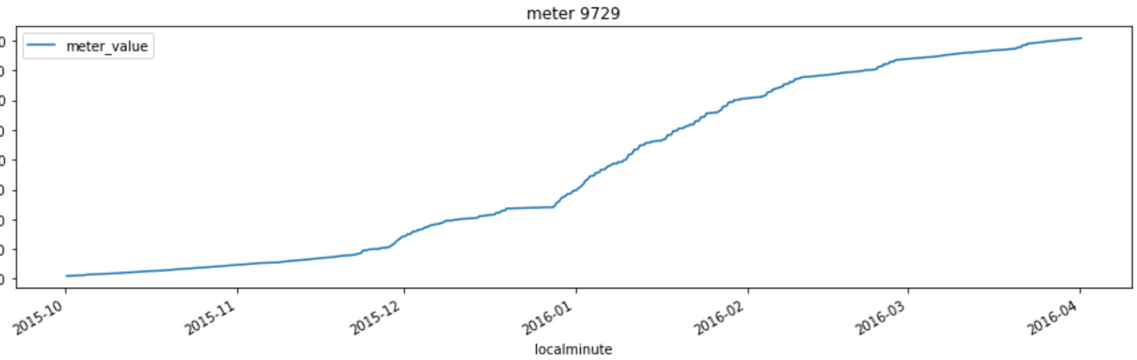

meter 9474
meter_value
235000
230000
225000
220000
215000
210000
205000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

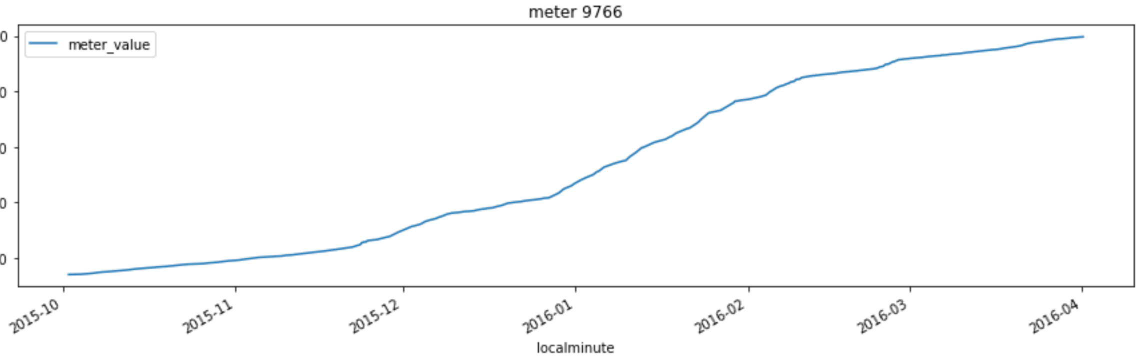

meter 9600
meter_value
124500
124000
123500
123000
122500
122000
121500
121000
2016-01-11
2016-01-13
2016-01-15
2016-01-17
2016-01-19
2016-01-21
2016-01-23
2016-01-25
2016-01-27
2016-01-29
localminute

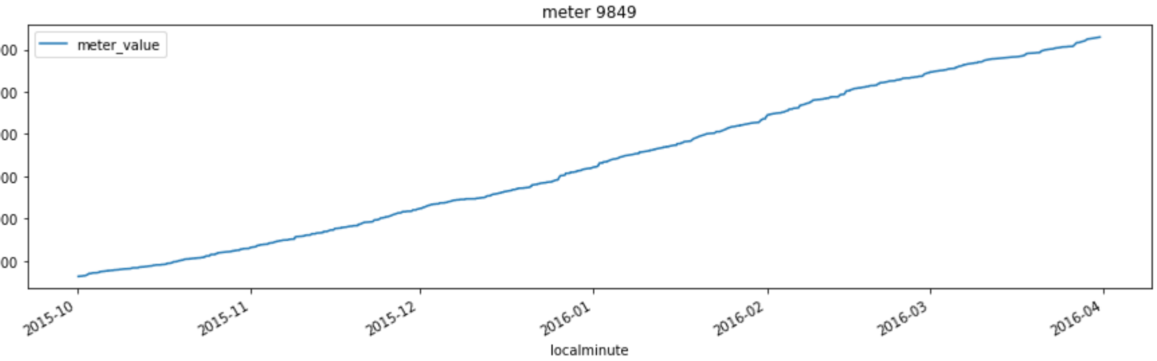

meter 9620
meter_value
465000
460000
455000
450000
445000
440000
435000
2015-11-11
2015-12-02
2015-12-23
2016-01-13
2016-02-03
2016-02-24
2016-03-16
localminute

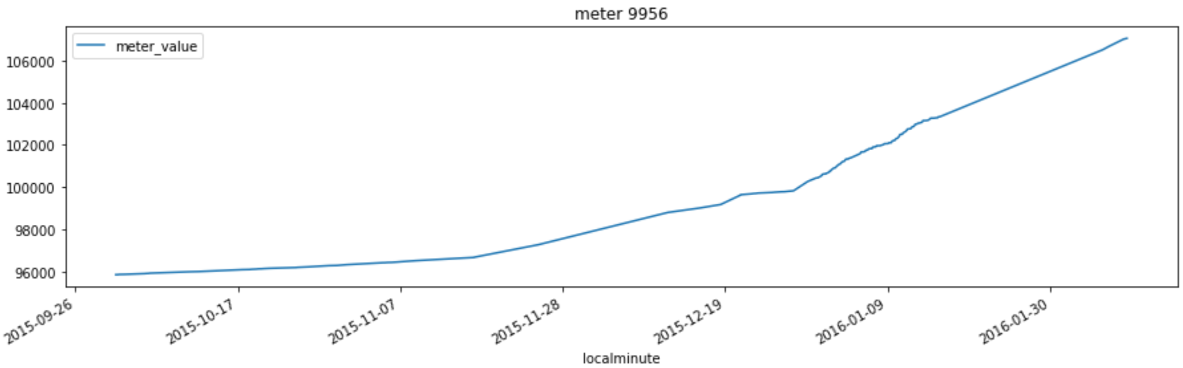

meter 9631
meter_value
124000
122000
120000
118000
116000
114000
112000
110000
108000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute

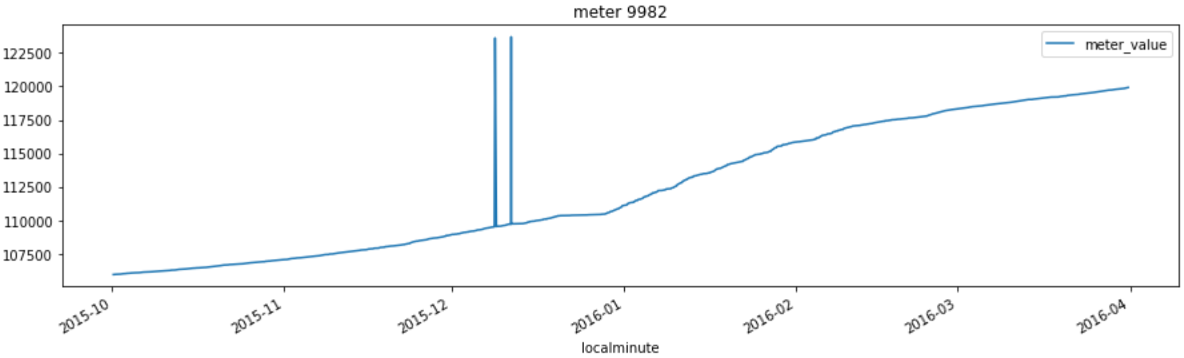

meter 9639
meter_value
250000
240000
230000
220000
210000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute


meter 9729
meter_value
138000
136000
134000
132000
130000
128000
126000
124000
122000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute


meter 9766
meter_value
180000
175000
170000
165000
160000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute


meter 9849
meter_value
39000
38000
37000
36000
35000
34000
2015-10
2015-11
2015-12
2016-01
2016-02
2016-03
2016-04
localminute





In [45]:

```
key = 8156
df_i = groups.get_group(key)
```

Select data by datetime period.

In [46]:

```
start_date = pd.to_datetime('2015-10-1')
end_date = pd.to_datetime('2015-10-31')
mask = (df_i.index > start_date) & (df_i.index <= end_date)

df_i_bymonth = df_i.iloc[mask]
display(df_i_bymonth.head())
len(df_i_bymonth)
```

	dataid	meter_value
localminute		
2015-10-01 05:02:37	8156	251818
2015-10-01 05:07:38	8156	251818
2015-10-01 05:44:37	8156	251818
2015-10-01 05:48:37	8156	251818
2015-10-01 06:09:37	8156	251818

Out[46]:

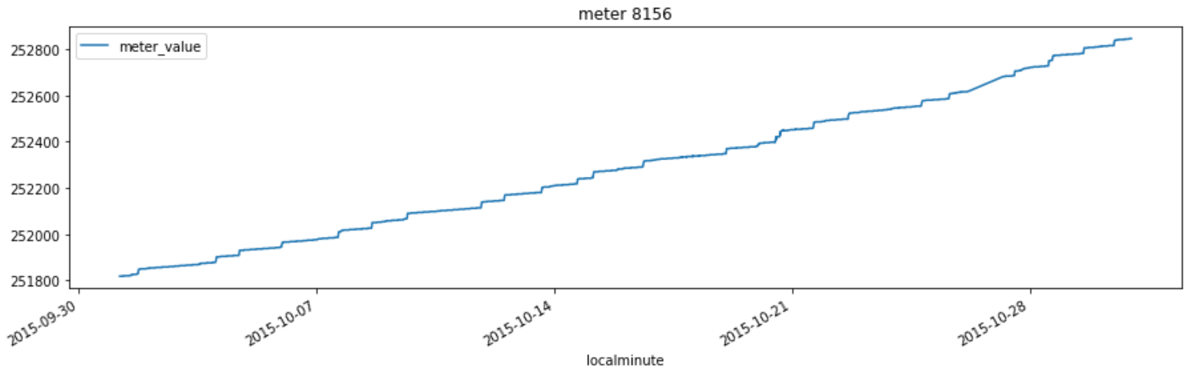
```
4475
```

In [47]:

```
df_i_bymonth.drop(columns='dataid').plot(figsize=(15,4), title=str(f'meter {key}'))
```

Out[47]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x16685de1d30>
```



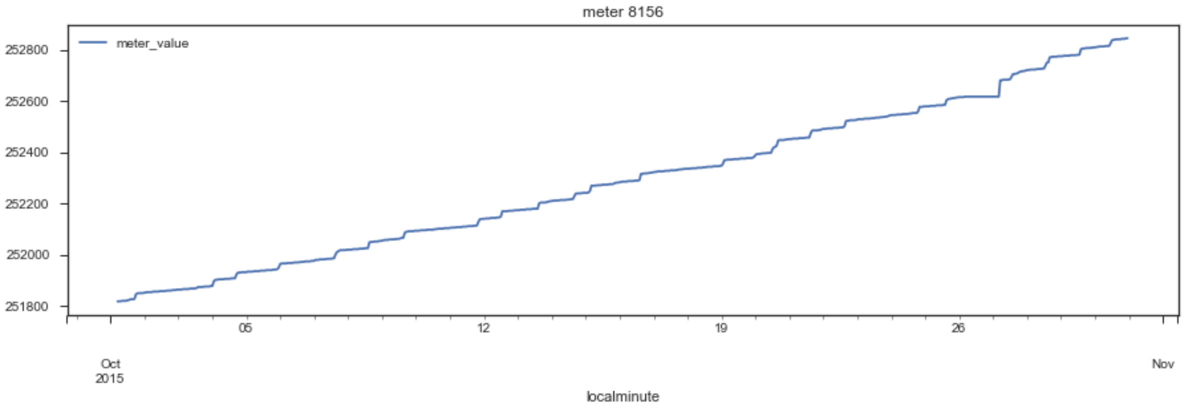
Resample with hourly frequency.

In [60]:

```
df_i_bymonth_2 = df_i_bymonth.drop(columns='dataid')
df_i_bymonth_2_resampled = df_i_bymonth_2.resample('H').mean().ffill()
df_i_bymonth_2_resampled.plot(figsize=(15,4), title=str(f'meter {key}'))
```

Out[60]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x16687548710>
```



In [52]:

```
display(len(df_i_bymonth_2_resampled))
```

```
715
```

In [53]:

```
# todo: clean dirty data from faulty meters by extrapolating the clean data.
```

In [ ]:

```

```