Exploratory Data Analysis for Charging Events

1. import data

I have imported and loaded the data as Pandas dataframe.

In [ ]:
```
import warnings
warnings.filterwarnings('ignore')#improve readability by disable warnings
```

In [ ]:
```
import pandas as pd
path = "data/Charging_events_data - charging_events_meter_reading.csv"
df = pd.read_csv(path)
df.head()
```

Out[ ]:

	Start Time	Meter Start (Wh)	Meter End(Wh)	Meter Total(Wh)	Total Duration (s)	Charger_name
0	24.08.2018 09:50	50	50.00	0.00	37	NaN
1	24.08.2018 09:51	50	50.00	0.00	38	NaN
2	24.08.2018 09:51	73	118.52	45.52	56	NaN
3	24.08.2018 09:53	105	116.66	11.66	76	NaN
4	24.08.2018 09:54	121	144.77	23.77	19	NaN

2. Preprocessing

In [ ]:
```
df.describe()
```

Out[ ]:

	Meter Start (Wh)	Meter End(Wh)	Meter Total(Wh)	Total Duration (s)
count	2.770000e+02	2.770000e+02	277.000000	2.770000e+02
mean	3.968875e+05	4.030848e+05	6197.316318	9.651005e+04
std	3.912772e+05	3.892371e+05	12260.182878	3.472706e+05
min	0.000000e+00	0.000000e+00	0.000000	0.000000e+00
25%	6.900900e+04	7.866592e+04	0.000000	1.200000e+01
50%	1.932000e+05	2.007288e+05	1380.280000	5.704000e+03
75%	7.430480e+05	7.508278e+05	6822.500000	7.343900e+04
max	1.204911e+06	1.204935e+06	126350.920000	3.020411e+06

According to the information provided by df.describe() above, we can observe that there are unnecessary rows in the dataset, e.g., rows where the Meter Total equals 0 and rows where the Total Duration equals 0. These rows are the situations where the "charging" did not happen, and the EV is not charged at all. Hence, we need to drop these rows. If we kept these rows, when we perform calculations relevant to "mean" or "median" for duration and meter total, the result would be biased.

Moreover, although 'Start Time' denotes the time that each charging event happened, it is in Object(string) format. Therefore, to facilitate the process for time-series analysis in the following steps, it would be better if we convert 'Start Time' to the date_time data type in Pandas and set it as our index.

In [ ]:
```
df_dropped_zero = df[(df['Total Duration (s)']!=0) & (df['Meter Total(Wh)']!
df_dropped_zero.loc[:,'Start Time'] = pd.to_datetime(df_dropped_zero['Start
df_dropped_zero.set_index('Start Time', inplace=True)#set as index
df_dropped_zero
```

Out[ ]:

Start Time	Meter Start (Wh)	Meter End(Wh)	Meter Total(Wh)	Total Duration (s)	Charger_name
2018-08-24 09:51:00	73	118.52	45.52	56	NaN
2018-08-24 09:53:00	105	116.66	11.66	76	NaN
2018-08-24 09:54:00	121	144.77	23.77	19	NaN
2018-08-27 09:16:00	39	483.18	444.18	515	NaN
2018-08-27 09:24:00	507	547.25	40.25	48	NaN
...	...	...	...	...	...
2019-09-09 05:47:00	0	33101.51	33101.51	6906	charger_1
2019-09-11 14:05:00	0	13807.38	13807.38	3726	charger_1
2019-09-12 11:05:00	0	35804.92	35804.92	7234	charger_1
2019-09-16 07:17:00	0	32996.70	32996.70	5240	charger_1
2019-09-16 09:33:00	0	17109.95	17109.95	4350	charger_1

175 rows × 5 columns

3. time analysis

dt_df: dataframe that only contains meter total and total duration, which are used in the following plots.

In [ ]:
```
dt_df = df_dropped_zero[['Meter Total(Wh)','Total Duration (s)']]
```

In [ ]:
```
agg_hour = dt_df.groupby(dt_df.index.hour).median() #group by hour, calculat
agg_hour['size'] = dt_df.groupby(dt_df.index.hour).size()#count frequency of
agg_hour
```

Out[ ]:

Start Time	Meter Total(Wh)	Total Duration (s)	size
3	3870.70	55140.0	1
4	4879.64	22586.0	1
5	9852.09	18944.0	5
6	7280.38	32632.0	11
7	5420.59	16860.0	15
8	1761.00	4886.0	17
9	687.03	2517.0	18
10	4273.80	19952.0	30
11	6518.61	70909.0	21
12	3790.13	14816.0	17
13	9958.04	84167.0	10
14	6823.34	61200.0	7
15	11852.42	67363.0	5
16	13679.24	57084.0	3
17	1568.59	49403.0	5
18	237.49	49740.0	3
20	11866.52	148896.5	2
21	405.52	293825.0	3
22	18834.29	12267.0	1

3.1 charging pattern by hour

in this section, I analyzed and plotted how charging patterns vary with different time in a day (different hours).

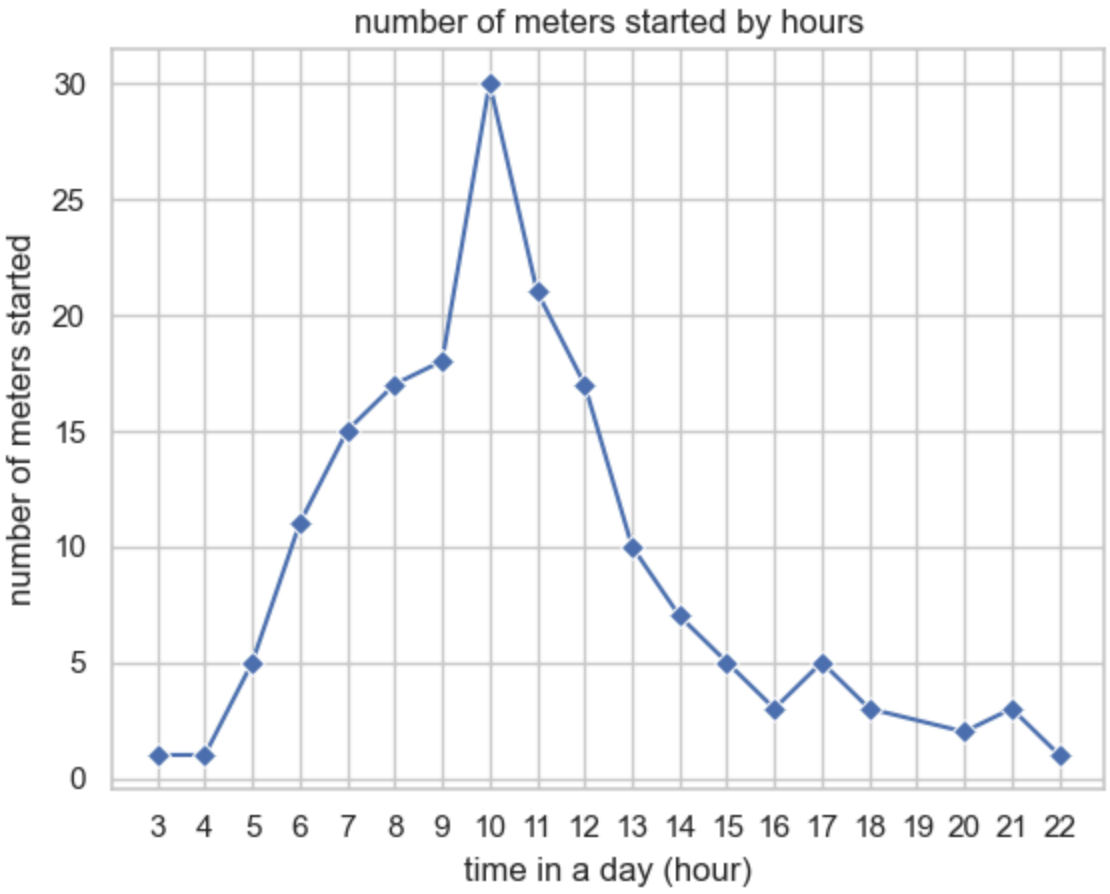
3.1.1

First, let's see the number of charging events in different hours in a day:

In [ ]:
```
import seaborn as sns
import matplotlib.pyplot as plt
sns.set(style = "whitegrid")
x_ticks = range(agg_hour.index.min(), agg_hour.index.max() + 1)
ax_ms_hours = sns.lineplot(data = agg_hour, x = agg_hour.index, y='size',mar
ax_ms_hours.set_xticks(x_ticks)
ax_ms_hours.set_xlabel("time in a day (hour)", fontsize=12, labelpad=5)
```

```
ax_ms_hours.set_ylabel("number of meters started", fontsize=12, labelpad=5)
ax_ms_hours.set_title("number of meters started by hours", fontsize=12)
```

Out[ ]: Text(0.5, 1.0, 'number of meters started by hours')



According to the plot above, we can tell that most of the charging events occur between 6:00 to 13:00, and the peak of # of charging events happens at 10:00. If our data is collected without bias, then we can conclude that people tend to charge their EV in the morning and in the noon of a day.

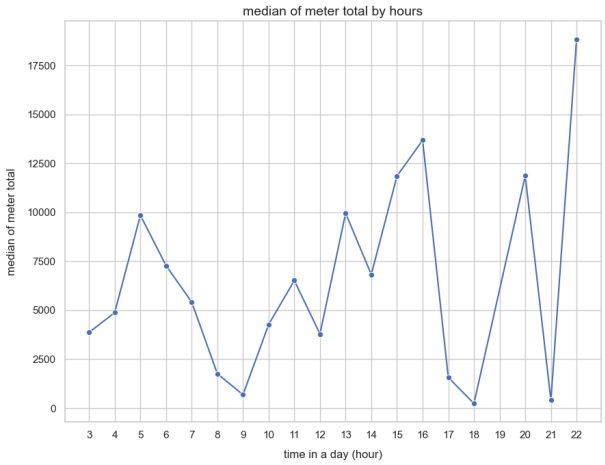
3.1.2

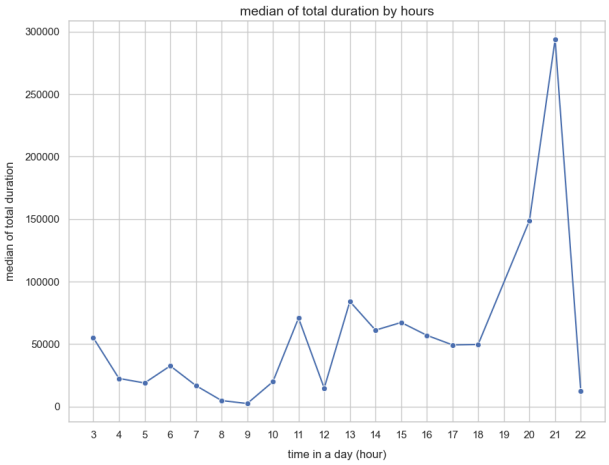
Then, I would like to see how the amount of energy charged and the duration of charging events vary in different times in a day. I chose to use the median for these two columns to observe the varying pattern , because median is affected by outliers less than mean.

In [ ]:
```
fig1, axes1 = plt.subplots(nrows=1, ncols=2, figsize=(18, 7))
sns.lineplot(ax = axes1[0], data = agg_hour, x = agg_hour.index, y='Meter To
axes1[0].set_xticks(x_ticks)
axes1[0].set_xlabel("time in a day (hour)", fontsize=12, labelpad=10)
axes1[0].set_ylabel("median of meter total", fontsize=12, labelpad=10)
axes1[0].set_title("median of meter total by hours", fontsize=14)

sns.lineplot(ax = axes1[1],data= agg_hour, x = agg_hour.index, y="Total Dura
axes1[1].set_xlabel("time in a day (hour)", fontsize=12, labelpad=10)
axes1[1].set_ylabel("median of total duration", fontsize=12, labelpad=10)
```

```
axes1[1].set_title("median of total duration by hours", fontsize=14)
axes1[1].set_xticks(x_ticks)
plt.tight_layout()
```





In the plots above, we can tell that both median of meter total by hours and median of total duration by hours fluctuates dramatically. This is probably because we only have 175 rows of data after dropping non-charging events, which is insufficient for observing an unbiased pattern of charging duration and the amount of energy charged by times in a day as we have 20 different hours recorded in a day in our data. There is no trends presented in the plots above.

3.2 Charging pattern by days in a week

In the following plots, we analyze how the charging pattern varies by different days in a week.

In [ ]:
```
agg_week = dt_df.groupby(dt_df.index.dayofweek).median()
agg_week['size'] = dt_df.groupby(dt_df.index.dayofweek).size()
agg_week.index = ['Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun']
agg_week
```

Out[ ]:

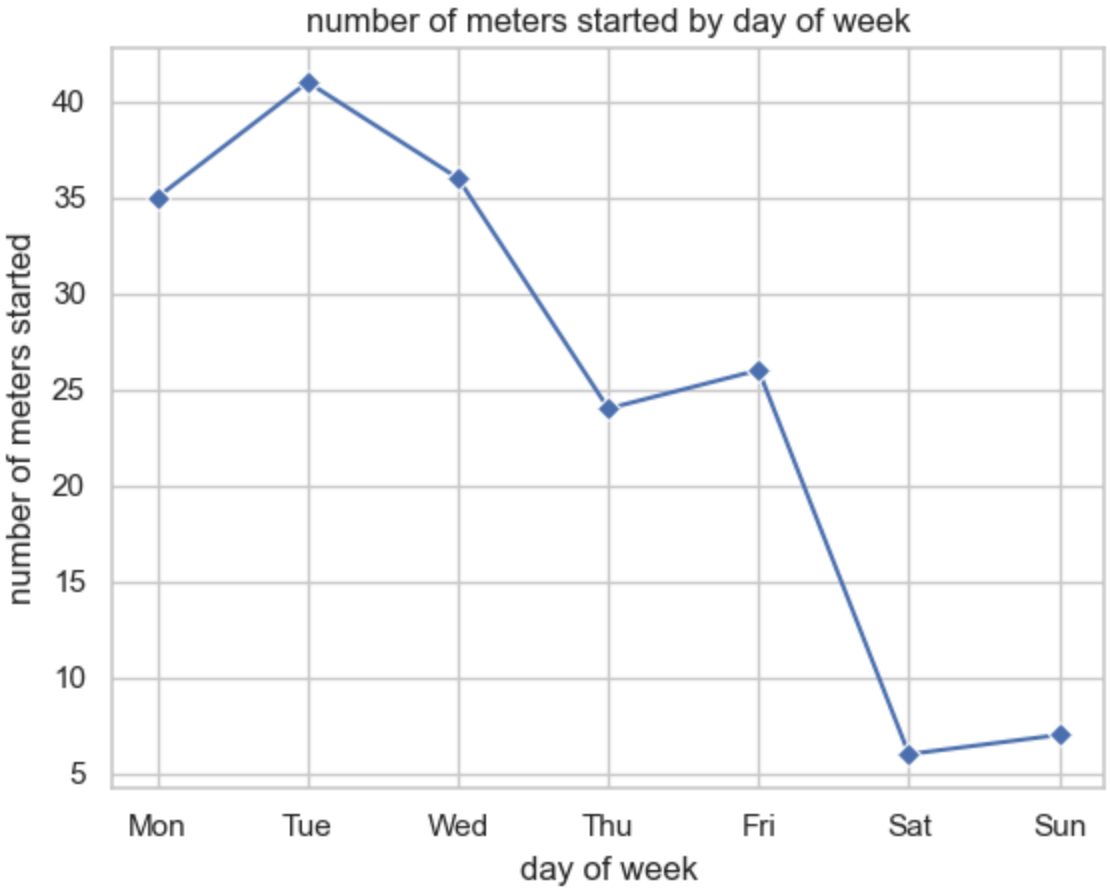
	Meter Total(Wh)	Total Duration (s)	size
Mon	5249.970	25574.0	35
Tue	2594.150	49403.0	41
Wed	3834.405	15353.5	36
Thu	6556.715	51772.0	24
Fri	6038.615	13587.0	26
Sat	4989.310	78103.5	6
Sun	5819.650	55140.0	7

3.2.1

First, I managed to plot the number of charging events (the number of meters started) vs. days in a week.

In [ ]:
```
ax_days = sns.lineplot(data = agg_week, x = agg_week.index, y='size',marker=
ax_days.set_xlabel("day of week", fontsize=12, labelpad=5)
ax_days.set_ylabel("number of meters started", fontsize=12, labelpad=5)
ax_days.set_title("number of meters started by day of week", fontsize=12)
```

Out[ ]:
```
Text(0.5, 1.0, 'number of meters started by day of week')
```



According to the plot above, people charge their EVs more during business days (Mon-Fri) than on weekends (Sat-Sun). Chances are people charge their vehicles to go to work (which happens on weekdays) more often than charging to go to travel (which happens on weekends). From what I know, charging EV is time consuming, so EVs are good for short-distance commute rather than traveling for long distance. Those people who own EVs may choose to use trains or so when traveling during weekends.

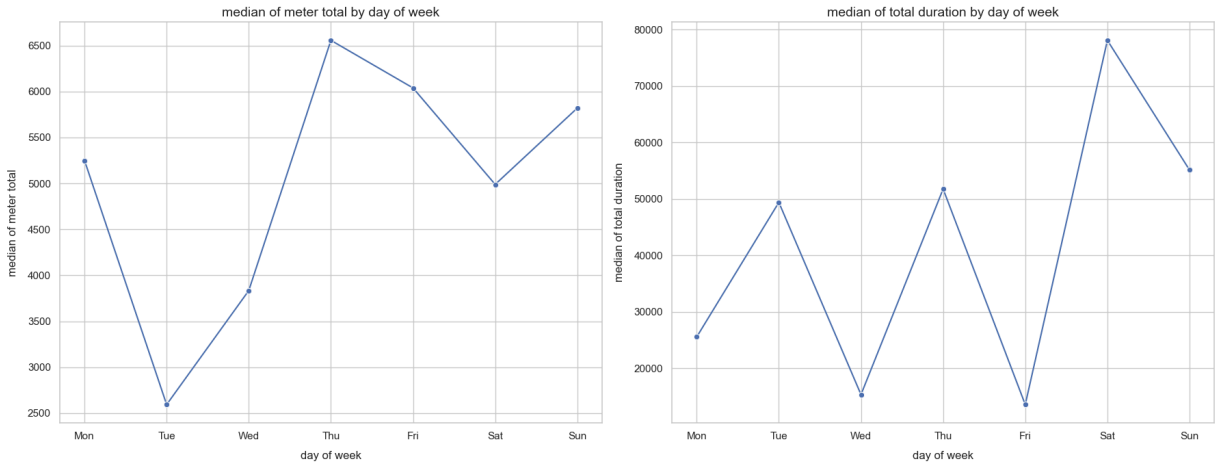
3.2.2

Next, I want to observe how the amount of energy charged and the charging duration vary by different days in a week.

In [ ]:
```
fig2, axes2 = plt.subplots(nrows=1, ncols=2, figsize=(18, 7))
sns.lineplot(ax = axes2[0], data = agg_week, x = agg_week.index, y='Meter To
axes2[0].set_xlabel("day of week", fontsize=12, labelpad=10)
```

```
axes2[0].set_ylabel("median of meter total", fontsize=12, labelpad=10)
axes2[0].set_title("median of meter total by day of week", fontsize=14)

sns.lineplot(ax = axes2[1],data= agg_week, x = agg_week.index, y="Total Dura
axes2[1].set_xlabel("day of week", fontsize=12, labelpad=10)
axes2[1].set_ylabel("median of total duration", fontsize=12, labelpad=10)
axes2[1].set_title("median of total duration by day of week", fontsize=14)
plt.tight_layout()
```



According to the plots above, we can tell that there is no observable trend in the graph —— just like the plots in 3.1.2, this is probably due to insufficient data.

3.3 Charging pattern overtime

In this section, we group the charging data by months and present each charging pattern overtime.

In [ ]:
```
all_time = dt_df.resample('M').median()
all_time['size'] = dt_df.resample('M').size()
all_time
```

Out[ ]:

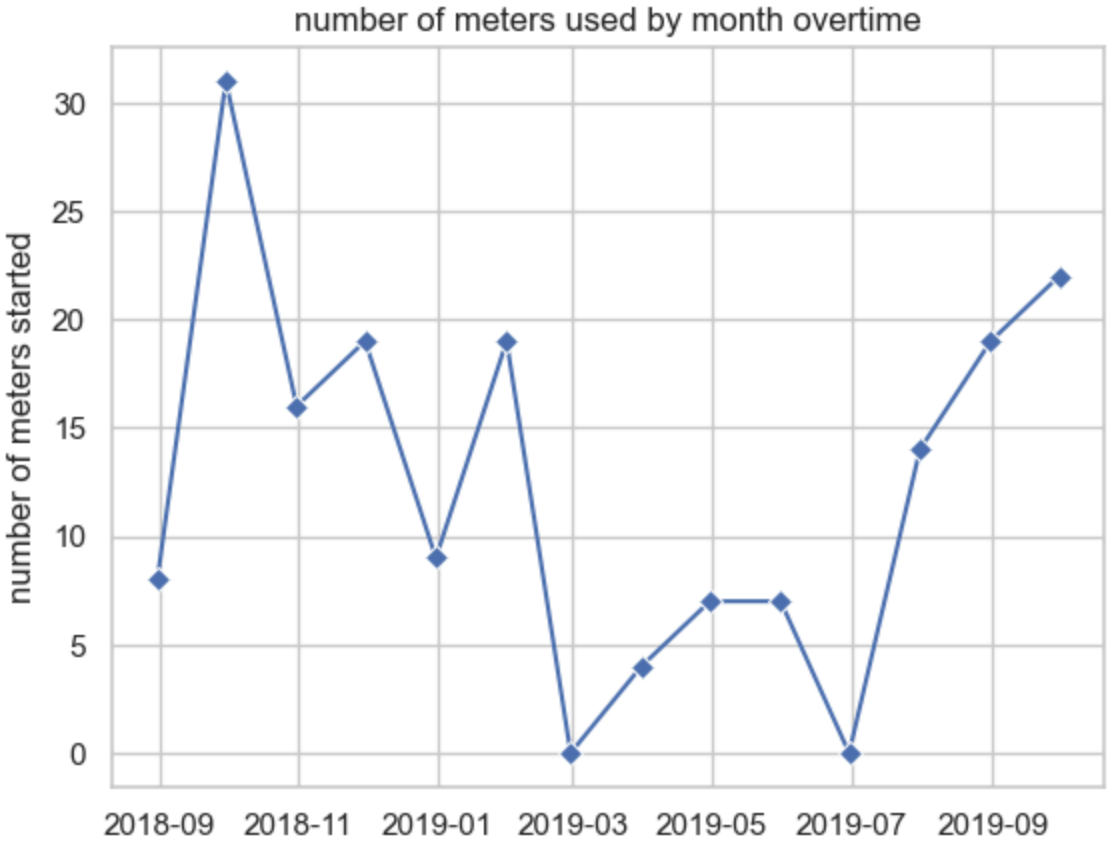
Start Time	Meter Total(Wh)	Total Duration (s)	size
2018-08-31	53.740	141.0	8
2018-09-30	2292.990	30877.0	31
2018-10-31	5630.460	78714.0	16
2018-11-30	6338.840	64064.0	19
2018-12-31	6558.650	78108.0	9
2019-01-31	7028.760	75360.0	19
2019-02-28	NaN	NaN	0
2019-03-31	5701.840	14180.5	4
2019-04-30	1133.160	6412.0	7
2019-05-31	351.090	49740.0	7
2019-06-30	NaN	NaN	0
2019-07-31	5141.590	21482.0	14
2019-08-31	4181.370	57084.0	19
2019-09-30	6493.955	7070.0	22

3.3.1

First, let's see how the number of charging events vary overtime:

In [ ]:
```
ax_days = sns.lineplot(data = all_time, x = all_time.index, y='size',marker=
ax_days.set_xlabel("")
ax_days.set_ylabel("number of meters started", fontsize=12, labelpad=5)
ax_days.set_title("number of meters used by month overtime", fontsize=12)
```

Out[ ]:
```
Text(0.5, 1.0, 'number of meters used by month overtime')
```



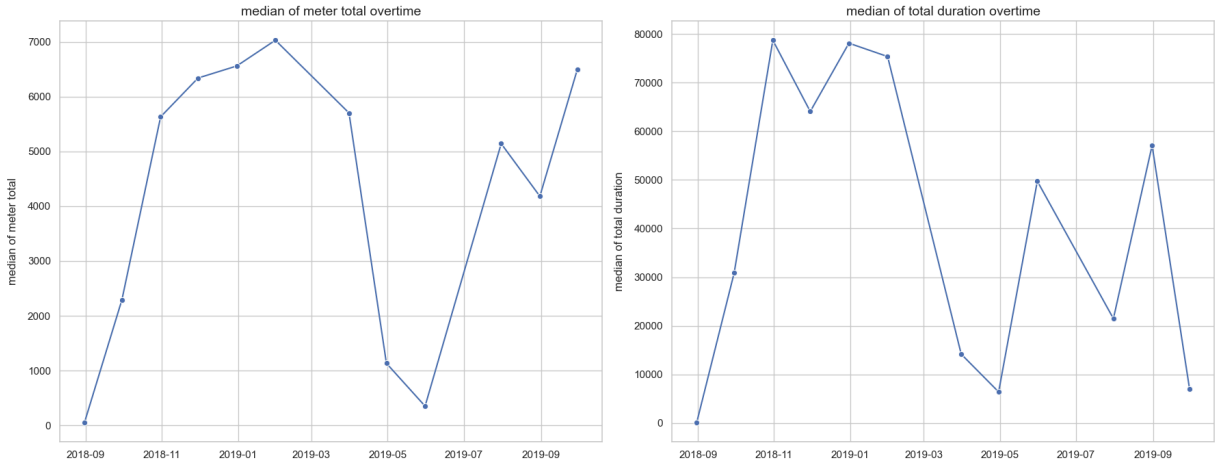
According to the plot above, there is a slump in terms of number of charging events between March 2019 to July 2019, and in Aug 2018. This indicates people tend to charge less during the summer.

3.3.2

Next, let's observe the charging pattern in terms of meter total and total duration overtime.

In [ ]:
```
fig3, axes3 = plt.subplots(nrows=1, ncols=2, figsize=(18, 7))
sns.lineplot(ax = axes3[0], data = all_time, x = all_time.index, y='Meter To
axes3[0].set_xlabel('')
axes3[0].set_ylabel("median of meter total", fontsize=12, labelpad=10)
axes3[0].set_title("median of meter total overtime", fontsize=14)

sns.lineplot(ax = axes3[1],data= all_time, x = all_time.index, y="Total Dura
axes3[1].set_xlabel('')
axes3[1].set_ylabel("median of total duration", fontsize=12, labelpad=10)
axes3[1].set_title("median of total duration overtime", fontsize=14)
plt.tight_layout()
```



According to the plots above, there are slumps during the summer of 2019, and in Aug 2018. In summary, people did not charge as much or as long as other time frames during the summer time. Chances are people are taking a summer break, e.g., going back to their hometown, so they temporarily do not use their EVs.

4. charger-wise analysis

In this section, I plotted and analyzed how charging behaviors vary by different chargers.

In [ ]:
```
agg_charger = df_dropped_zero.groupby('Charger_name').agg(
    median_duration = ('Total Duration (s)','median'),
    median_meter_total = ('Meter Total(Wh)','median')
)
agg_charger['size'] = df_dropped_zero.groupby('Charger_name').size()
agg_charger=agg_charger.reset_index(names='charger')
agg_charger
```

Out[ ]:

	charger	median_duration	median_meter_total	size
0	charger_1	4795.0	25053.325	6
1	charger_10	89578.0	1702.080	5
2	charger_11	22779.0	2594.150	9
3	charger_12	158899.0	15913.700	3
4	charger_13	22137.0	1618.530	3
5	charger_14	15097.0	2475.670	7
6	charger_15	73515.0	5249.970	5
7	charger_16	5123.0	12059.190	3
8	charger_2	33893.5	5047.710	22
9	charger_3	32408.0	2292.990	27
10	charger_4	49740.0	4698.860	43
11	charger_5	75360.0	1663.640	9
12	charger_6	74715.0	4804.890	4
13	charger_7	22586.0	4879.640	1
14	charger_8	64064.0	6558.650	19
15	charger_9	823282.0	6785.630	1

In [ ]:
```
#function for sorting all chargers in order
import re
def natural_sort_key(s):
    return [int(text) if text.isdigit() else text.lower() for text in re.spl
sorted_chargers = sorted(agg_charger['charger'].unique(),key=natural_sort_ke
```

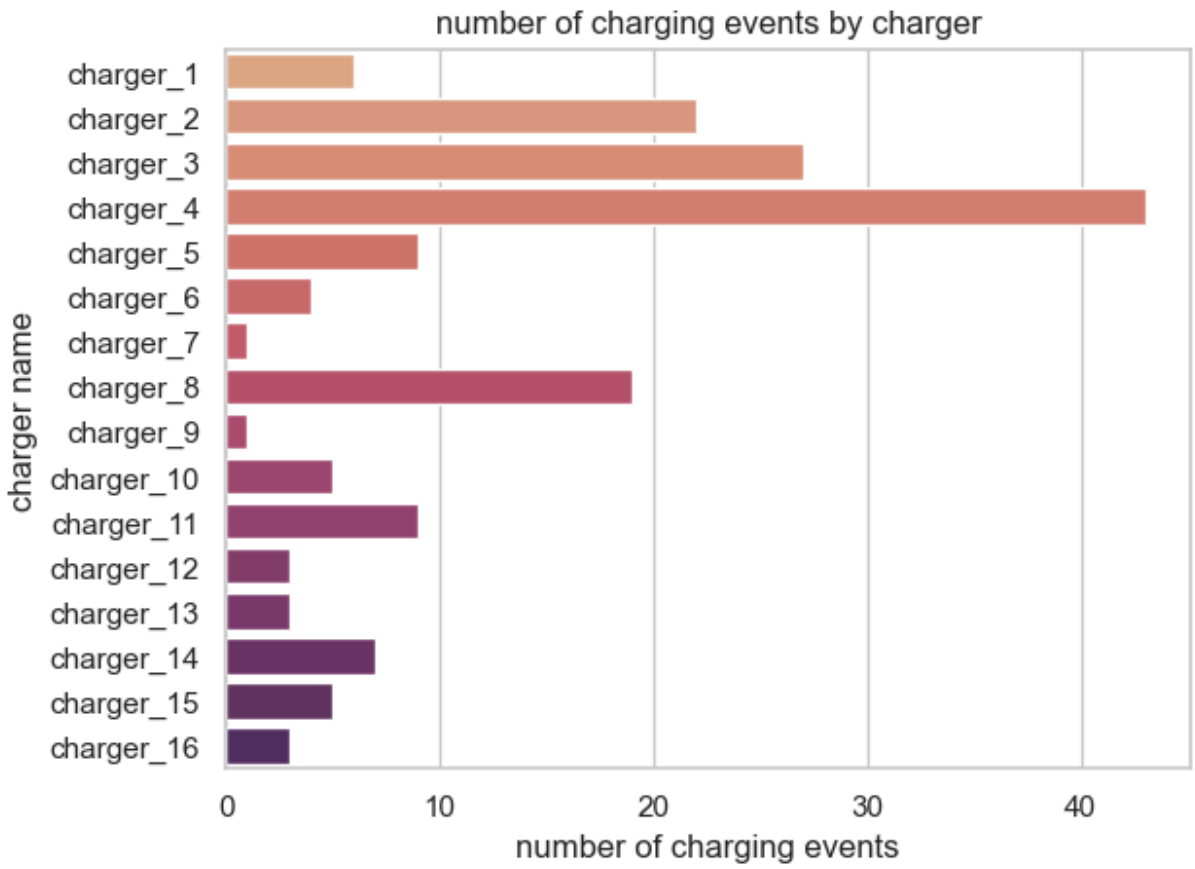
4.1 Charging pattern by chargers

4.1.1

First, I plotted how the number of charging events differ by chargers

In [ ]:
```
ax_size_charger = sns.barplot(x='size',y='charger',data=agg_charger,palette=
ax_size_charger.set_xlabel("number of charging events")
ax_size_charger.set_ylabel("charger name")
ax_size_charger.set_title("number of charging events by charger")
```

Out[ ]:
```
Text(0.5, 1.0, 'number of charging events by charger')
```



From the plot above, we can tell that charger 7 and charger 9 are barely used, while charger 2/3/4/8 are used most often. This may due to the location those chargers are placed and their distinct capabilities of charging (e.g. how fast can it charge, how much energy can it charge at one, how expensive does charging cost).

4.1.2

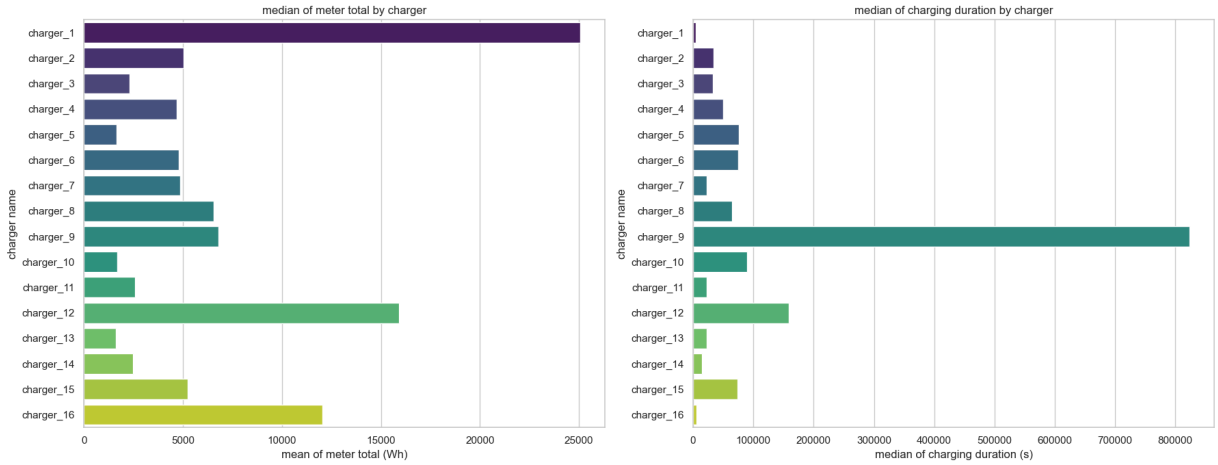
Then, I plotted how the median of charging duration and the amount of energy charged (meter total) vary by different chargers..

In [ ]:
```
fig4, axes4 = plt.subplots(nrows=1, ncols=2, figsize=(18, 7))

sns.barplot(x='median_meter_total', y='charger', data=agg_charger, ax=axes4[
axes4[0].set_title('median of meter total by charger')
axes4[0].set_xlabel('mean of meter total (Wh)')
axes4[0].set_ylabel('charger name')

sns.barplot(x='median_duration', y='charger', data=agg_charger, ax=axes4[1],
axes4[1].set_title('median of charging duration by charger')
axes4[1].set_xlabel('median of charging duration (s)')
axes4[1].set_ylabel('charger name')


plt.tight_layout()
```



According to the box plot above, people tend to charge a lot amount of energy (Wh) at charger 1/12/16, and people tend to charge a long time at charger 9. The big difference between these chargers shown on graph may due to the insufficiency of data, because for some chargers there are only one or two recorded data, which may cause our result relevant to them overly high or low. We need to also plot the boxplot to gain more insights into the data.

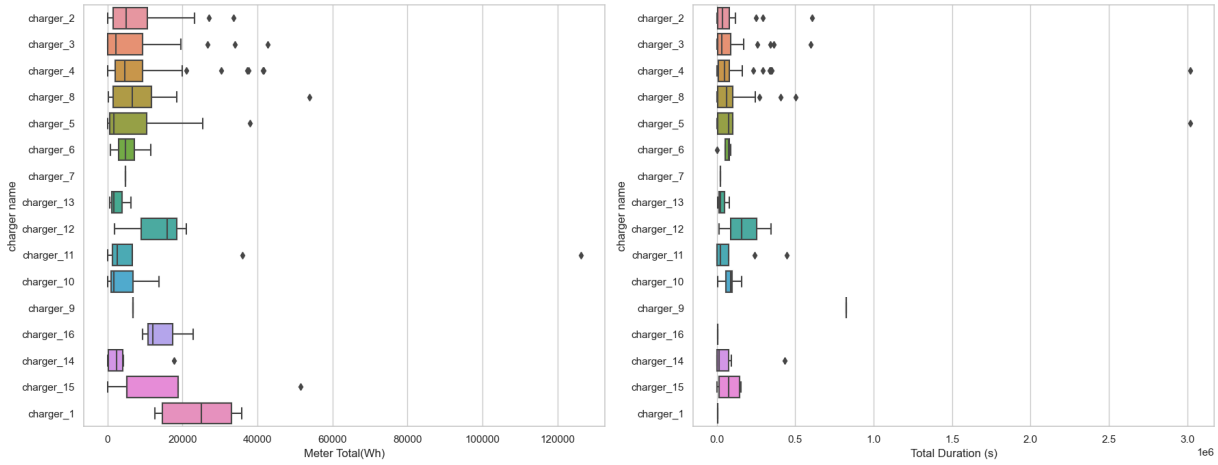
4.1.3

I have thus plotted boxplots for the chargers in terms of their median of energy charged (meter total) and charging duration.

In [ ]:
```
fig5, axes5 = plt.subplots(nrows=1, ncols=2, figsize=(18, 7))
sns.boxplot(ax=axes5[0], data=df_dropped_zero, x="Meter Total(Wh)", y="Charg
axes5[0].set_ylabel('charger name')

sns.boxplot(ax=axes5[1], data=df_dropped_zero, x="Total Duration (s)", y="Ch
axes5[1].set_ylabel('charger name')

plt.tight_layout()
```



According to the boxplots above, considering both the median, the minimum, the maximum, and lower as well as upper quartile value, we can tell that people tend to

charge the most amount of energy at Charger 1. Again, this is probably due to the location of capability of chargers.

5. Correlation analysis

In this section, I have used correlation analysis, basically scatter plot and regression line, to analyze if my intuitions for EV charging events are correct.

5.1 Does charging for longer time means more energy charged?

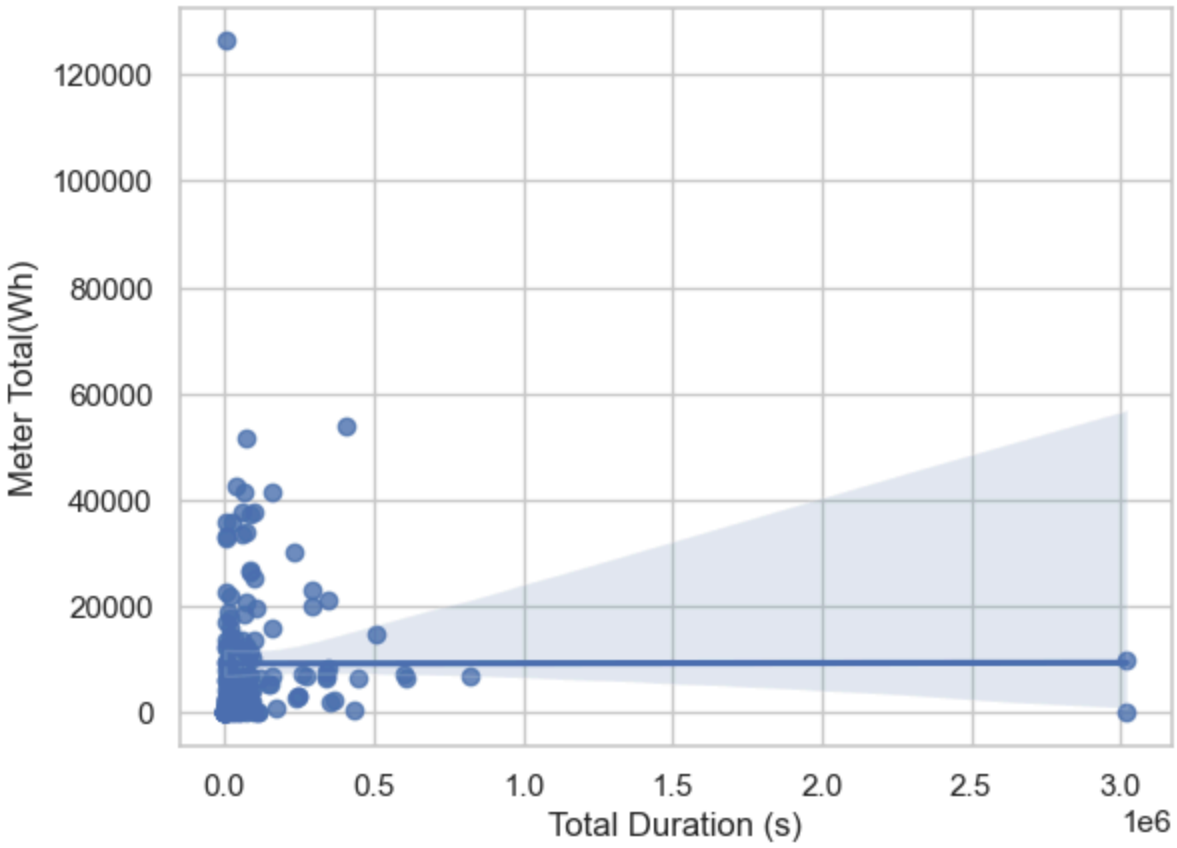
In this section, I want to figure out if charging for longer time means more energy charged. To be more specific, this question is about "is Total Duration (s) and Meter Total (wh) positively correlated". By my intuition, longer charging time should indicate more energy being charged.

5.1.1

First, we plot a scatter plot and the regression line for total duration and meter total.

In [ ]:
```
sns.regplot(data = df_dropped_zero, x="Total Duration (s)", y="Meter Total(W
```

Out[ ]:
```
<Axes: xlabel='Total Duration (s)', ylabel='Meter Total(Wh)'>
```



However, from the above plot, we can tell that the data we have is highly skewed, and some extreme values are affecting our regression line. Thus, we need to perform a log transformation on every data records so that the extreme values are diminished.

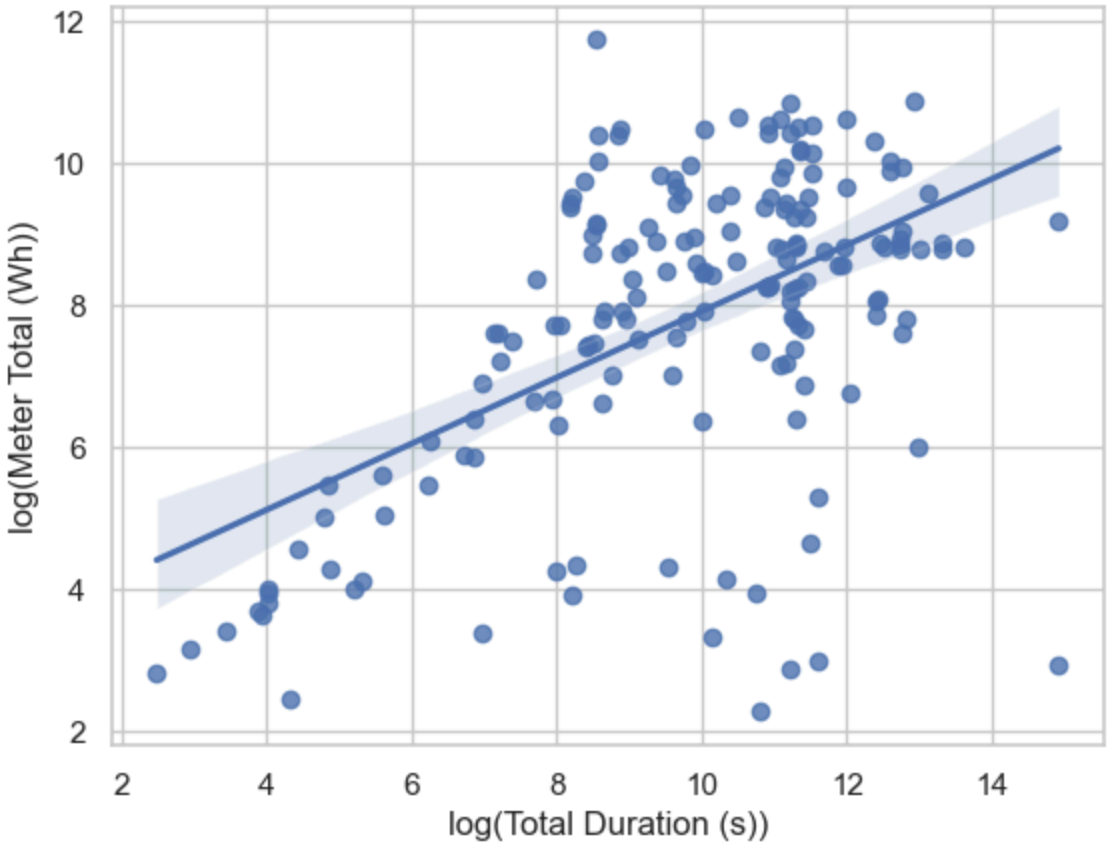
In [ ]:
```
import numpy as np
#perform log transformation
logged_dur = np.log(df_dropped_zero['Total Duration (s)'])
logged_meter = np.log(df_dropped_zero['Meter Total(Wh)'])
```

5.1.2

Then, we plot the scatter plot and the regression line again using log-transformed data.

In [ ]:
```
ax_dur_total = sns.regplot(x=logged_dur, y=logged_meter)
ax_dur_total.set_xlabel("log(Total Duration (s))")
ax_dur_total.set_ylabel("log(Meter Total (Wh))")
```

Out[ ]:
```
Text(0, 0.5, 'log(Meter Total (Wh))')
```



Now the graph above looks much nicer. According to the regression line and the scatter points, we can conclude that there is a positive correlation between total duration, i.e., the charging time, and meter total, i.e., the energy you have charged. Hence, my intuition is correct —— longer charging time indicates more energy charged.

5.2 Does lower meter reading when you begin to charge indicate that you are going to charge for longer?

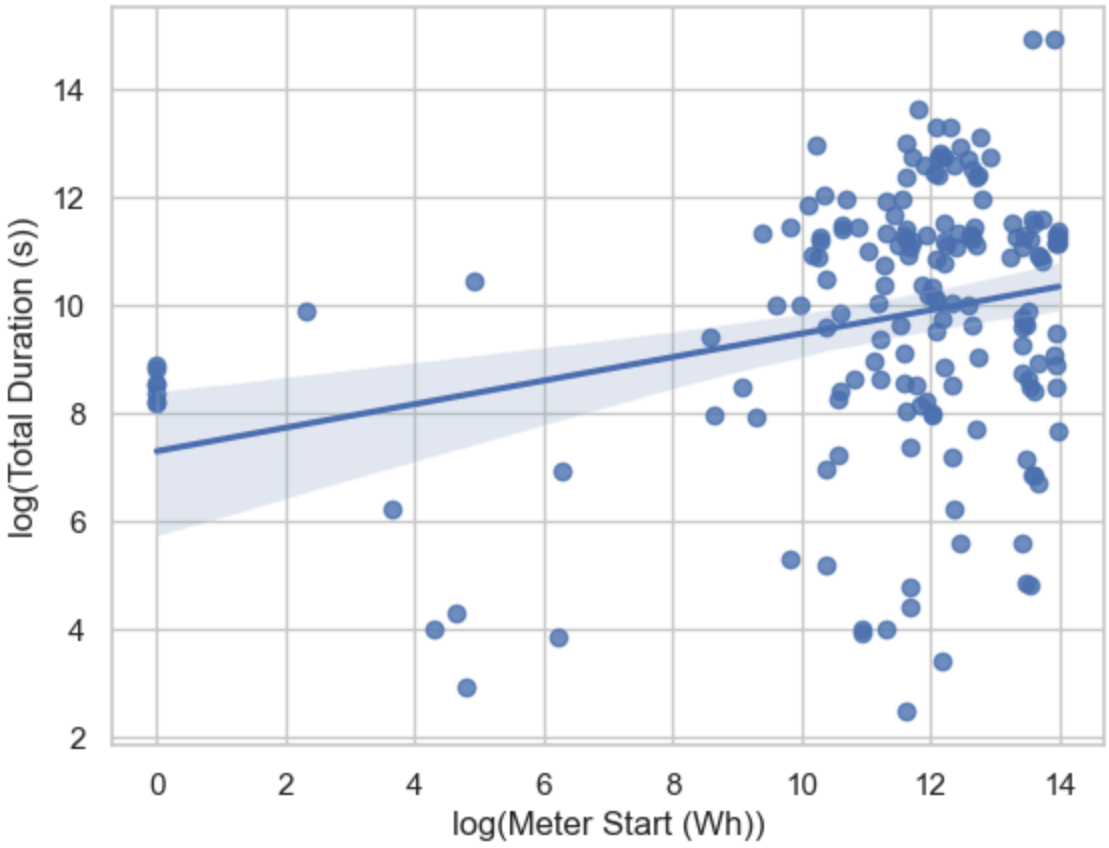
By intuition, if I start from a low meter reading, I will leave my EV for a longer time at the charger to ensure it is fully charged. Hence, I want to figure out if lower meter reading is related to the charging duration, i.e., is meter start negatively correlated to total duration?

5.2.1

We draw the scatter plot and regression line for meter start and total duration after log transformation.

In [ ]:
```
logged_start = [np.log(x) if x!=0 else 0 for x in df_dropped_zero['Meter Sta
ax_start_dur=sns.regplot(x=logged_start, y=logged_dur)
ax_start_dur.set_xlabel("log(Meter Start (Wh))")
ax_start_dur.set_ylabel("log(Total Duration (s))")
```

Out[ ]:
```
Text(0, 0.5, 'log(Total Duration (s))')
```



According to the graph above, there is no negative correlation between meter start and total duration. In contrast, there is a positive correlation, meaning higher meter reading when people begin to charge indicates that they are going to charge for a longer time. This is not in line with my intuition. I reckon this is because different EVs have huge differences between their capabilities of charging, so the "high meter reading" for some

EV is actually their "low meter reading", thus they still need a long time to be charged full.

6. Conclusion

According to my analysis, during a day, people tend to charge in the morning or in the noon; during a week, people tend to charge more during business days rather than on weekends; during a year, people tend to charge less in summer than in other seasons. In terms of chargers, each charger is used very differently, and this may due to the charger's capability and location. By the correlation analysis, there is a positive correlation betwen charging duration and the amount of energy charged, and there is a positive correlation between the meter reading people begin to charge and the charging duration.

Limitations:

I have plotted many graphs, but it is obvious that the data records, which are 175 rows after dropping non-charging events, are a little bit insufficient for a very comprehensive, unbiased report. For example, some chargers only have one record, which may cause its relevant data to be overly high or low.