EV_Charging

April 28, 2024

1 Exploratory Data Analysis on EV Charging

Author: Yuxuan Chen

Date: 04/28/2024

1.1 Introduction

This analysis focuses on exploring the patterns of electric vehicle (EV) charging data to understand usage trends and identify factors that influence energy consumption. By analyzing the temporal distribution of meter readings, we aim to discern when and how users interact with charging stations. This insight is crucial for optimizing energy resource management and preparing for peak demand times.

[146]:
```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
from scipy.stats import pearsonr
```

1.2 Basic Information

[147]:
```
charging_data = pd.read_csv('meter_reading.csv')
print(charging_data.dtypes)
print(charging_data.info())
charging_data.describe()
```

```
Start Time              object
Meter Start (Wh)         int64
Meter End(Wh)          float64
Meter Total(Wh)        float64
Total Duration (s)       int64
Charger_name            object
dtype: object
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 277 entries, 0 to 276
Data columns (total 6 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Start Time          277 non-null    object
```

```
 1   Meter Start (Wh)    277 non-null    int64
 2   Meter End(Wh)       277 non-null    float64
 3   Meter Total(Wh)     277 non-null    float64
 4   Total Duration (s)  277 non-null    int64
 5   Charger_name        264 non-null    object
dtypes: float64(2), int64(2), object(2)
memory usage: 13.1+ KB
None
```

[147]:

	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

After looking at the brief information and structure of dataset. The dataset consists of 277 entries across four primary features. Notably, the 'charger_name' column contains 13 null values. Given that these null entries constitute only a small fraction of the dataset, we have decided to drop these rows for the purposes of further analysis. Additionally, the 'start time' column is currently formatted as an object type; we plan to convert this to a time series type to facilitate more precise temporal analysis.

[148]:

```
charging_data = charging_data.dropna()
charging_data['Start Time'] = pd.to_datetime(charging_data['Start Time'],␣
 ↪format='%d.%m.%Y %H:%M')
charging_data.info()
charging_data.describe()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 264 entries, 9 to 276
Data columns (total 6 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Start Time          264 non-null    datetime64[ns]
 1   Meter Start (Wh)    264 non-null    int64
 2   Meter End(Wh)       264 non-null    float64
 3   Meter Total(Wh)     264 non-null    float64
 4   Total Duration (s)  264 non-null    int64
 5   Charger_name        264 non-null    object
dtypes: datetime64[ns](1), float64(2), int64(2), object(1)
memory usage: 14.4+ KB
```

[148]:

	Meter Start (Wh)	Meter End(Wh)	Meter Total(Wh)	Total Duration (s)
count	2.640000e+02	2.640000e+02	264.000000	2.640000e+02

mean	4.142639e+05	4.206191e+05	6355.157538	9.341541e+04
std	3.912358e+05	3.889345e+05	12429.480078	3.353374e+05
min	0.000000e+00	0.000000e+00	0.000000	0.000000e+00
25%	8.107250e+04	9.921000e+04	0.000000	0.000000e+00
50%	2.073220e+05	2.239153e+05	1682.860000	7.003500e+03
75%	7.532460e+05	7.532460e+05	6860.370000	7.392875e+04
max	1.204911e+06	1.204935e+06	126350.920000	3.020411e+06

[149]:
```
calculated_total_meter = charging_data['Meter End(Wh)'] - charging_data['Meter␣
↪Start (Wh)']
check_total_meter = charging_data['Meter Total(Wh)'] == calculated_total_meter
equality_check = check_total_meter.value_counts()
equality_check_summary
```

[149]:
```
True    264
Name: Check Equality, dtype: int64
```

Upon reviewing the dataset for any typographical errors or inconsistencies, we confirmed that there are no errors present. Everything appears to be in order, allowing us to proceed with the analysis without concerns about data accuracy or integrity.

[150]:
```
correlation, _ = pearsonr(charging_data['Total Duration (s)'],␣
↪charging_data['Meter Total(Wh)'])
print(f'The Pearson correlation coefficient between total duration and total␣
↪meter readings is: {correlation:.2f}')
```

```
The Pearson correlation coefficient between total duration and total meter
readings is: 0.03
```

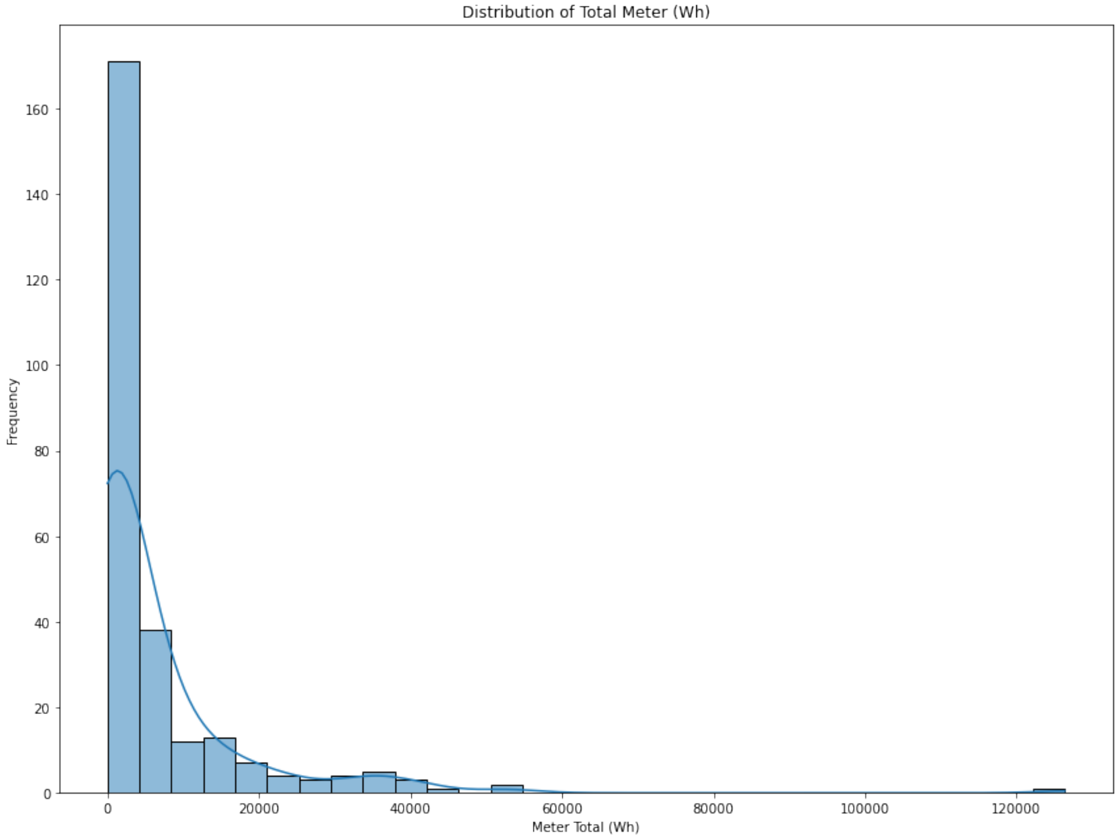
A Pearson correlation coefficient of 0.03 indicates a very weak positive linear relationship between the two variables. In practical terms, this means that there is essentially no meaningful linear correlation between total duration and total meter readings. Changes in one variable do not predict changes in the other in a significant way.

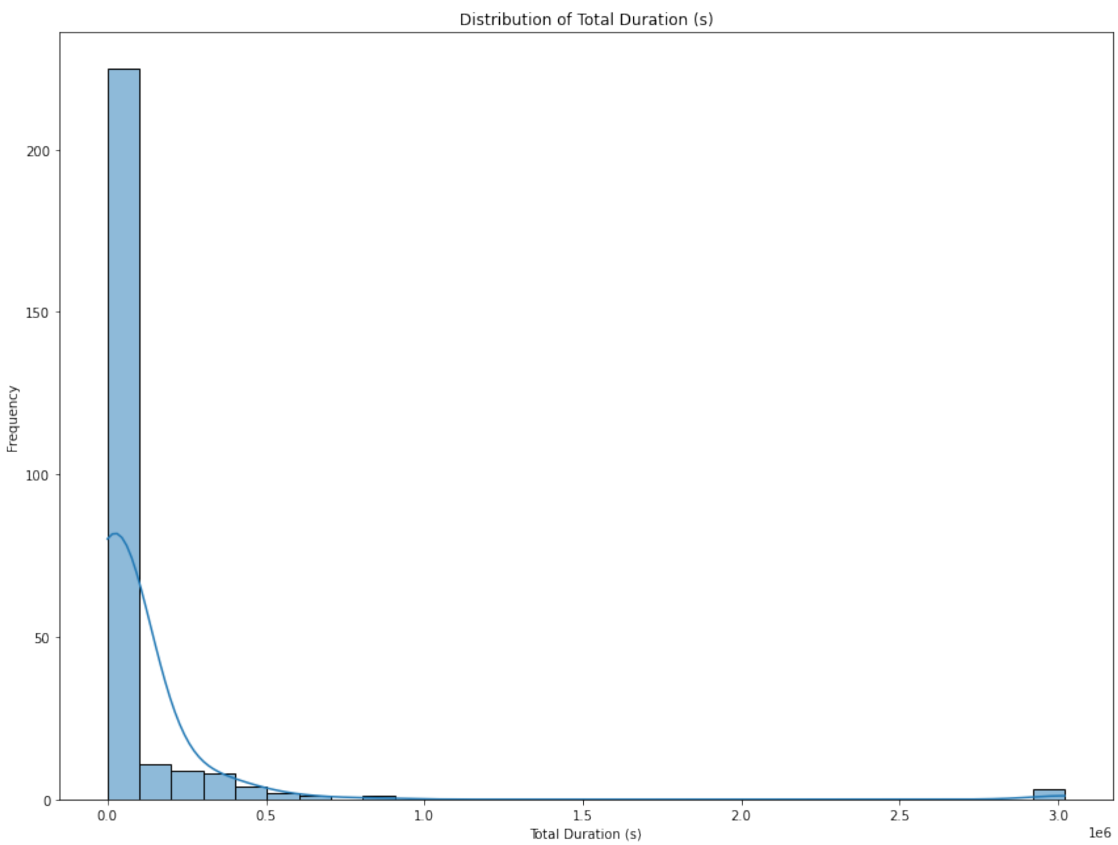
1.3 Investigating key features

Next, we choose to plot the distribution of key variables incluing total meter, total duration. We want to know how the data looks like visually.

[151]:
```
fig, axs = plt.subplots(2, 1, figsize=(12, 18))

sns.histplot(charging_data['Meter Total(Wh)'], bins=30, kde=True, ax=axs[0])
axs[0].set_title('Distribution of Total Meter (Wh)')
axs[0].set_xlabel('Meter Total (Wh)')
axs[0].set_ylabel('Frequency')

sns.histplot(charging_data['Total Duration (s)'], bins=30, kde=True, ax=axs[1])
axs[1].set_title('Distribution of Total Duration (s)')
axs[1].set_xlabel('Total Duration (s)')
```

```
axs[1].set_ylabel('Frequency')


plt.tight_layout()
plt.show()
```


Distribution of Total Meter (Wh)
Frequency
Meter Total (Wh)
0
20
40
60
80
100
120
140
160
0
20000
40000
60000
80000
100000
120000


Distribution of Total Duration (s)
Frequency
Total Duration (s)
0
50
100
150
200
0.0
0.5
1.0
1.5
2.0
2.5
3.0
1e6

After reviewing the graphs, we've noticed an unusual number of zeros in both the total meter and total duration graphs, which is irregular. One possible explanation for both the total duration and meter total being zero is that there was no charging activity and the charger was not used. However, if situations arise where the total meter is not zero but the total duration is—or vice versa—it could indicate serious issues. For example, this could suggest potential energy theft at the charger station or a malfunction in the time tracking system. Let's investigate further to confirm these suspicions.

1.4 Anomalous Detection

[152]:
```
anomalous_cases_1 = charging_data[(charging_data['Total Duration (s)'] == 0) &␣
↪(charging_data['Meter Total(Wh)'] > 0)]
anomalous_cases_2 = charging_data[(charging_data['Meter Total(Wh)'] == 0) &␣
↪(charging_data['Total Duration (s)'] > 0)]
anomalous = pd.concat([anomalous_cases_1,anomalous_cases_2],ignore_index= True)
anomalous
```

[152]:

	Start Time	Meter Start (Wh)	Meter End(Wh)	Meter Total(Wh)	\
0	2018-08-29 08:01:00	1546	2290.81	744.81	
1	2018-08-29 08:19:00	2433	2447.41	14.41	
2	2018-08-29 13:25:00	2441	18233.89	15792.89	
3	2018-09-06 07:27:00	5709	11062.20	5353.20	
4	2018-09-06 09:15:00	28619	31060.46	2441.46	
5	2018-12-19 10:55:00	622842	626820.04	3978.04	
6	2019-04-02 09:48:00	726247	727279.73	1032.73	
7	2019-04-04 10:27:00	675539	675826.10	287.10	
8	2019-01-28 01:11:00	873775	875702.62	1927.62	
9	2019-05-02 11:24:00	876263	876299.00	36.00	
10	2019-01-08 13:50:00	876296	876346.55	50.55	
11	2019-07-02 18:46:00	1131830	1139717.21	7887.21	
12	2019-08-15 12:48:00	1188459	1193044.31	4585.31	
13	2019-08-22 10:45:00	1197207	1202457.29	5250.29	
14	2019-01-08 13:01:00	1202446	1204934.59	2488.59	
15	2019-07-08 16:32:00	108481	110209.37	1728.37	
16	2019-08-26 11:12:00	126346	131199.08	4853.08	
17	2019-08-29 15:04:00	131185	134244.89	3059.89	
18	2019-09-04 09:40:00	134241	134258.79	17.79	
19	2019-09-05 07:49:00	42760	44466.08	1706.08	
20	2019-09-10 08:44:00	134241	134241.34	0.34	
21	2019-09-26 09:54:00	141023	141844.65	821.65	
22	2018-08-31 09:41:00	18263	18263.00	0.00	
23	2018-09-07 11:06:00	31051	31051.00	0.00	
24	2018-09-07 11:09:00	31051	31051.00	0.00	
25	2018-09-25 20:47:00	99210	99210.00	0.00	
26	2018-09-25 21:37:00	99210	99210.00	0.00	
27	2018-09-28 06:48:00	113105	113105.00	0.00	

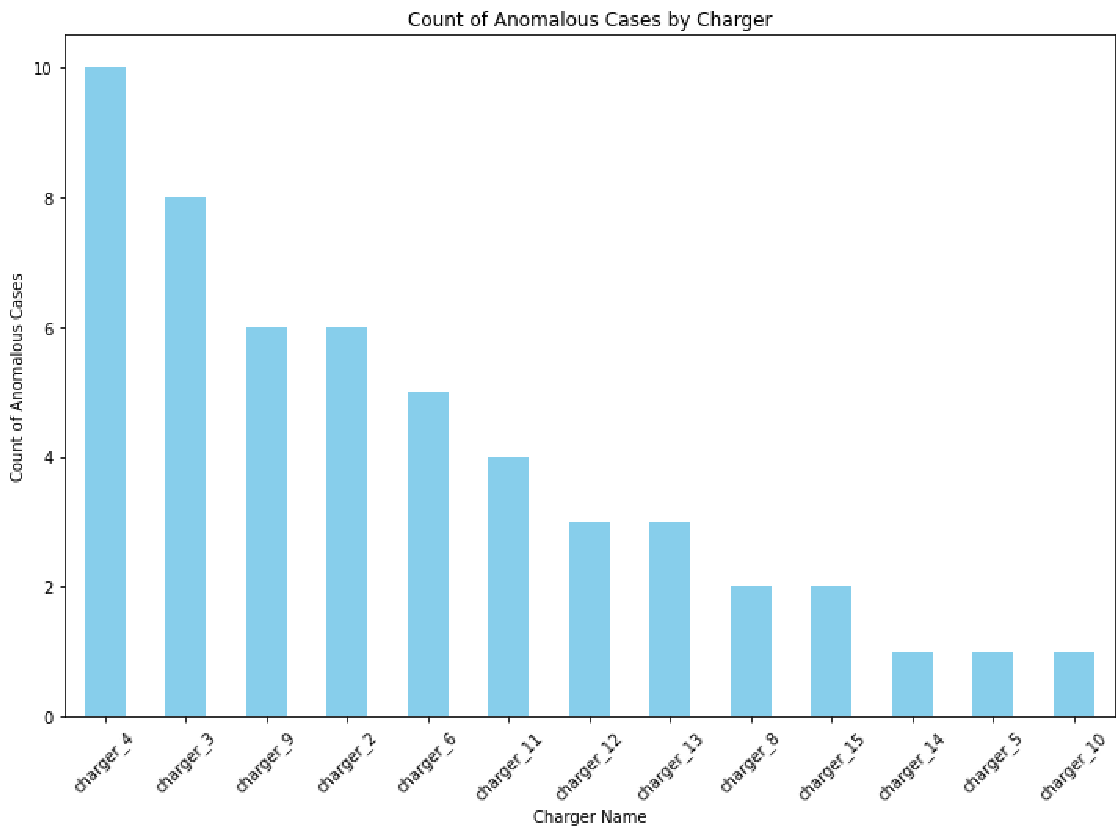
28	2018-09-28 07:39:00	113106	113106.00	0.00
29	2018-10-05 07:07:00	113106	113106.00	0.00
30	2018-10-18 10:20:00	193200	193200.00	0.00
31	2018-12-12 07:59:00	336215	336215.00	0.00
32	2019-03-25 11:26:00	672669	672669.00	0.00
33	2019-03-26 10:42:00	682489	682489.00	0.00
34	2019-03-26 10:46:00	682489	682489.00	0.00
35	2019-01-08 13:01:00	1139700	1139700.00	0.00
36	2019-07-03 08:00:00	1139700	1139700.00	0.00
37	2019-08-04 09:50:00	1181849	1181849.00	0.00
38	2019-01-08 13:01:00	1204910	1204910.00	0.00
39	2019-06-27 09:29:00	0	0.00	0.00
40	2019-06-28 13:05:00	0	0.00	0.00
41	2019-07-03 13:31:00	0	0.00	0.00
42	2019-07-04 15:28:00	0	0.00	0.00
43	2019-07-04 14:00:00	0	0.00	0.00
44	2019-09-12 04:58:00	15700	15700.00	0.00
45	2019-08-21 08:16:00	120731	120731.00	0.00
46	2019-08-21 08:25:00	120800	120800.00	0.00
47	2019-07-02 12:32:00	131185	131185.00	0.00
48	2019-09-26 06:46:00	141023	141023.00	0.00
49	2019-08-28 12:50:00	56590	56590.00	0.00
50	2019-09-05 07:23:00	81050	81050.00	0.00
51	2019-09-23 12:40:00	81050	81050.00	0.00

	Total Duration (s)	Charger_name
0	0	charger_3
1	0	charger_3
2	0	charger_3
3	0	charger_2
4	0	charger_3
5	0	charger_5
6	0	charger_4
7	0	charger_8
8	0	charger_4
9	0	charger_4
10	0	charger_4
11	0	charger_4
12	0	charger_6
13	0	charger_6
14	0	charger_6
15	0	charger_11
16	0	charger_11
17	0	charger_9
18	0	charger_9
19	0	charger_10
20	0	charger_9

```
21                  0     charger_9
22                 55     charger_3
23                150     charger_3
24              10566     charger_3
25               3015     charger_2
26              36118     charger_2
27               3003     charger_2
28                 81     charger_2
29                 12     charger_2
30                 12     charger_3
31              40947     charger_8
32               7313     charger_4
33                211     charger_4
34                 51     charger_4
35            3020396     charger_4
36              94534     charger_4
37                 24     charger_6
38              10005     charger_6
39              93211    charger_13
40             248320    charger_12
41              71978    charger_12
42             398097    charger_12
43             415981    charger_13
44               4200    charger_13
45                 11    charger_11
46                 29    charger_11
47              64730     charger_9
48               6713     charger_9
49              85411    charger_14
50              14206    charger_15
51                 14    charger_15
```

[153]:

```
charger_counts = anomalous['Charger_name'].value_counts()
plt.figure(figsize=(12, 8))
charger_counts.plot(kind='bar', color='skyblue')
plt.title('Count of Anomalous Cases by Charger')
plt.xlabel('Charger Name')
plt.ylabel('Count of Anomalous Cases')
plt.xticks(rotation=45)
plt.show()
```



After extracting all the data samples with zeros in either the meter total or total duration, we found that 57 entries exhibit this issue. Moreover, among the 15 chargers, 13 have reported such discrepancies. Notably, Charger 4 has the highest number of these cases. If these anomalies aren't due to typos or operational errors, the charger company should be notified to investigate the matter further. This could help in identifying any underlying issues with the equipment or possible misuse.

[154]:
```
condition = (charging_data['Total Duration (s)'] == 0) & (charging_data['Meter␣
↪Total(Wh)'] > 0)
condition_2 = (charging_data['Total Duration (s)'] >0) & (charging_data['Meter␣
↪Total(Wh)'] == 0)
charging_data = charging_data.drop(charging_data[condition].index)
charging_data = charging_data.drop(charging_data[condition_2].index)

correlation, _ = pearsonr(charging_data['Total Duration (s)'],␣
↪charging_data['Meter Total(Wh)'])
print(f'The Pearson correlation coefficient between total duration and total␣
↪meter readings is: {correlation:.2f}')
```

```
The Pearson correlation coefficient between total duration and total meter
readings is: 0.04
```

```
<ipython-input-154-2339a68eca91>:4: UserWarning: Boolean Series key will be
```

```
reindexed to match DataFrame index.
  charging_data = charging_data.drop(charging_data[condition_2].index)
```

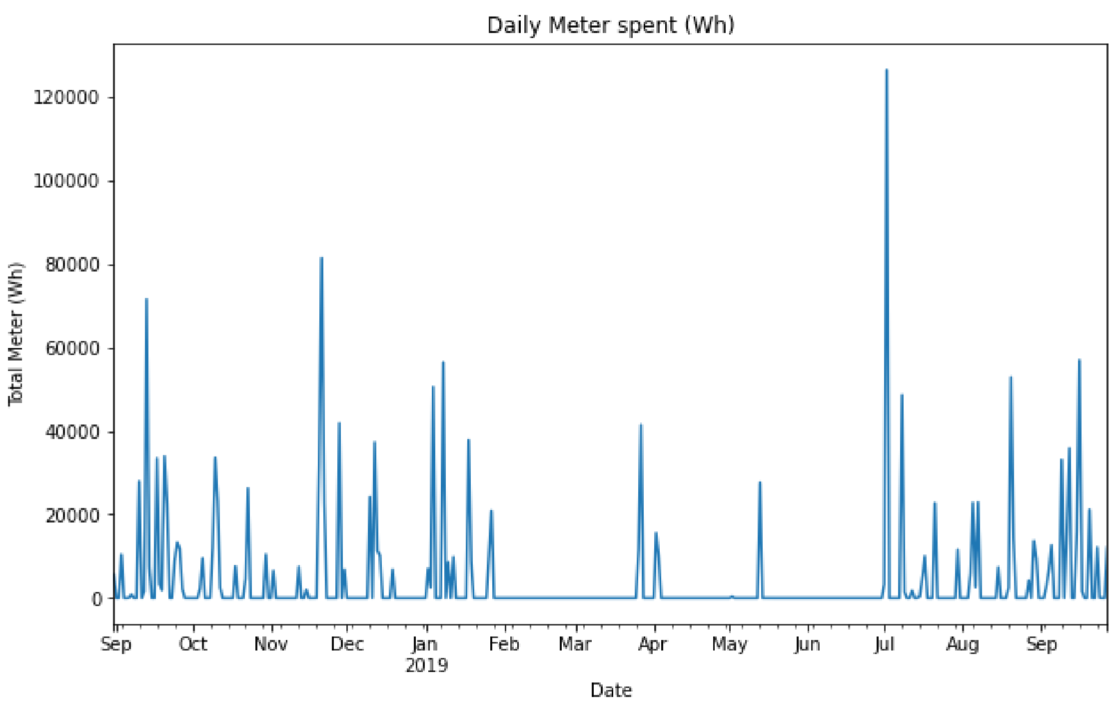
Dropping the anomalous data from charging_data and then recalculating the Pearson correlation coefficient, which increased slightly to 0.04, does indicate a marginal improvement in the correlation between total duration and total meter readings. However, as you noted, this correlation remains very weak.

1.5 Meter Total by week, day, hour

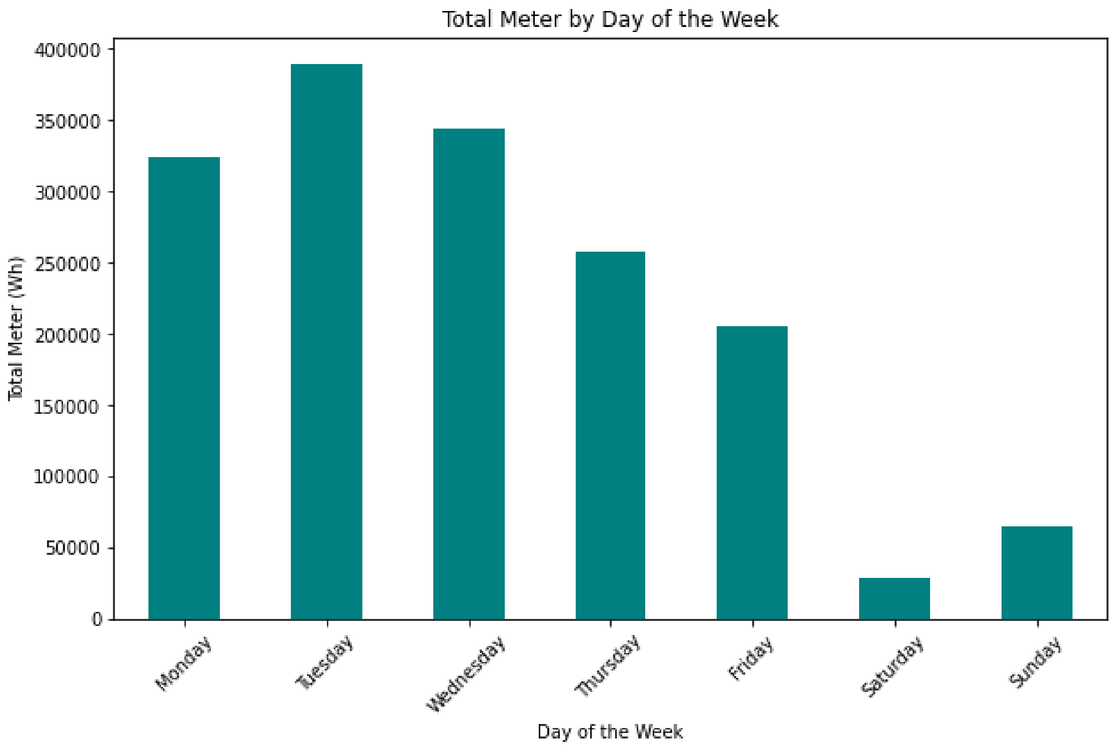
Next, we want to delve into the relationship between energy consumption and time, we'll visualize the distribution of total meter readings across different time segments: weekly, daily, and hourly. By plotting these distributions, we aim to uncover any recurring patterns. These could manifest as particular days of the week with higher usage, daily peaks and troughs, or specific hours that consistently show increased activity. Such insights could be pivotal in optimizing energy management and anticipating demand.

[155]:
```
plt.figure(figsize=(10, 6))
charging_data.set_index('Start Time', inplace=True)
charging_data['Meter Total(Wh)'].resample('D').sum().plot()
plt.title('Daily Meter spent (Wh)')
plt.xlabel('Date')
plt.ylabel('Total Meter (Wh)')
```

[155]: Text(0, 0.5, 'Total Meter (Wh)')

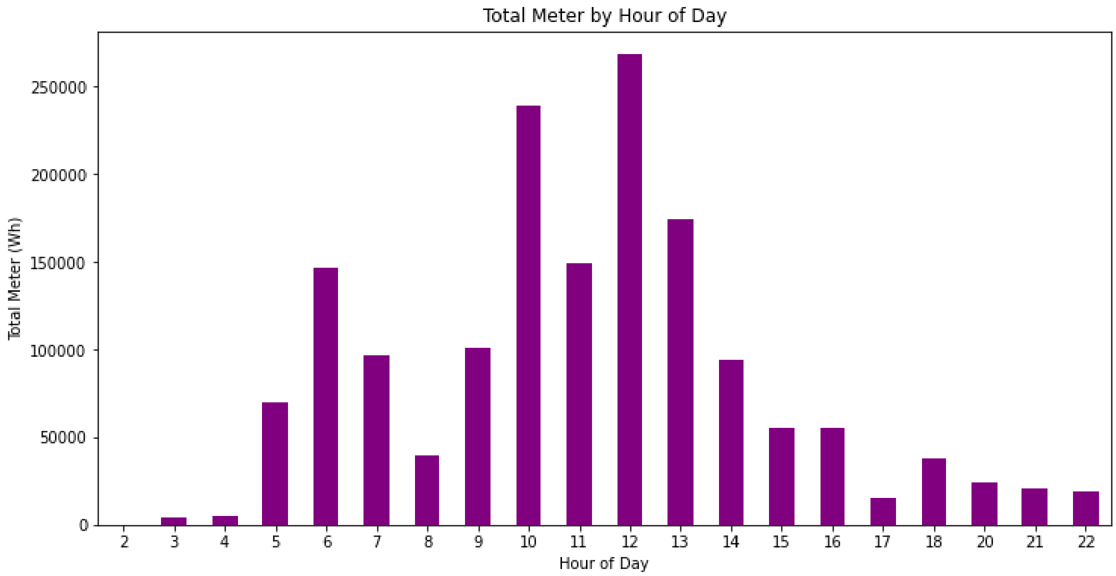


[156]:
```
charging_data.reset_index(inplace=True)
charging_data['Day of Week'] = charging_data['Start Time'].dt.day_name()

weekly_energy = charging_data.groupby('Day of Week')['Meter Total(Wh)'].sum().
↪reindex([
    'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday'
])

plt.figure(figsize=(10, 6))
weekly_energy.plot(kind='bar', color='teal')
plt.title('Total Meter by Day of the Week')
plt.xlabel('Day of the Week')
plt.ylabel('Total Meter (Wh)')
plt.xticks(rotation=45)
plt.show()
```



[157]:
```
charging_data['Hour of Day'] = charging_data['Start Time'].dt.hour
hourly_total_energy = charging_data.groupby('Hour of Day')['Meter Total(Wh)'].
↪sum()

plt.figure(figsize=(12, 6))
hourly_total_energy.plot(kind='bar', color='purple')
plt.title('Total Meter by Hour of Day')
```

```
plt.xlabel('Hour of Day')
plt.ylabel('Total Meter (Wh)')
plt.xticks(rotation=0)
plt.show()
```



The plots reveal that energy consumption reaches its peak during midday on weekdays, suggesting a surge in usage as activities ramp up. Conversely, the lowest levels of consumption occur around midnight on weekends, which could reflect a significant drop in demand when most activities wind down. These findings highlight key periods of high and low energy usage that could inform strategies for managing load and efficiency.

1.6 Summary

The investigation commenced with a diligent data preparation phase, where we addressed missing values and reformatted the 'Start Time' column to enable precise time series analysis. Our exploratory data analysis (EDA) yielded critical insights into the charging habits over different time scales, shedding light on daily and hourly consumption peaks and troughs.

During the EDA, we encountered an array of anomalous data points characterized by discrepancies in the duration and energy meter readings. Specifically, instances were found where charging sessions had significant energy consumption recorded but zero duration logged, and vice versa. This raised concerns over potential underlying issues such as system malfunctions or unauthorized energy usage, which could lead to operational inefficiencies or revenue losses.

After cleaning the dataset of these anomalies, we re-evaluated the correlation between the total charging duration and energy consumed, which remained notably weak. This suggested that the charging duration is not a sole predictor of energy usage, indicating the influence of other, possibly more complex, factors at play.

The analysis culminated in the visualization of energy consumption patterns, revealing a peak in usage around noon on weekdays, juxtaposing with the lowest usage around midnight during weekends. Such patterns point towards a workweek-centric charging behavior.

The discovery of anomalous data is a critical finding, emphasizing the need for the charging company to implement stringent monitoring systems and potentially revise their operational protocols. Addressing these issues is not only crucial for maintaining the reliability and accuracy of the charging service but also for safeguarding against energy theft and ensuring the integrity of usage data for future analyses.