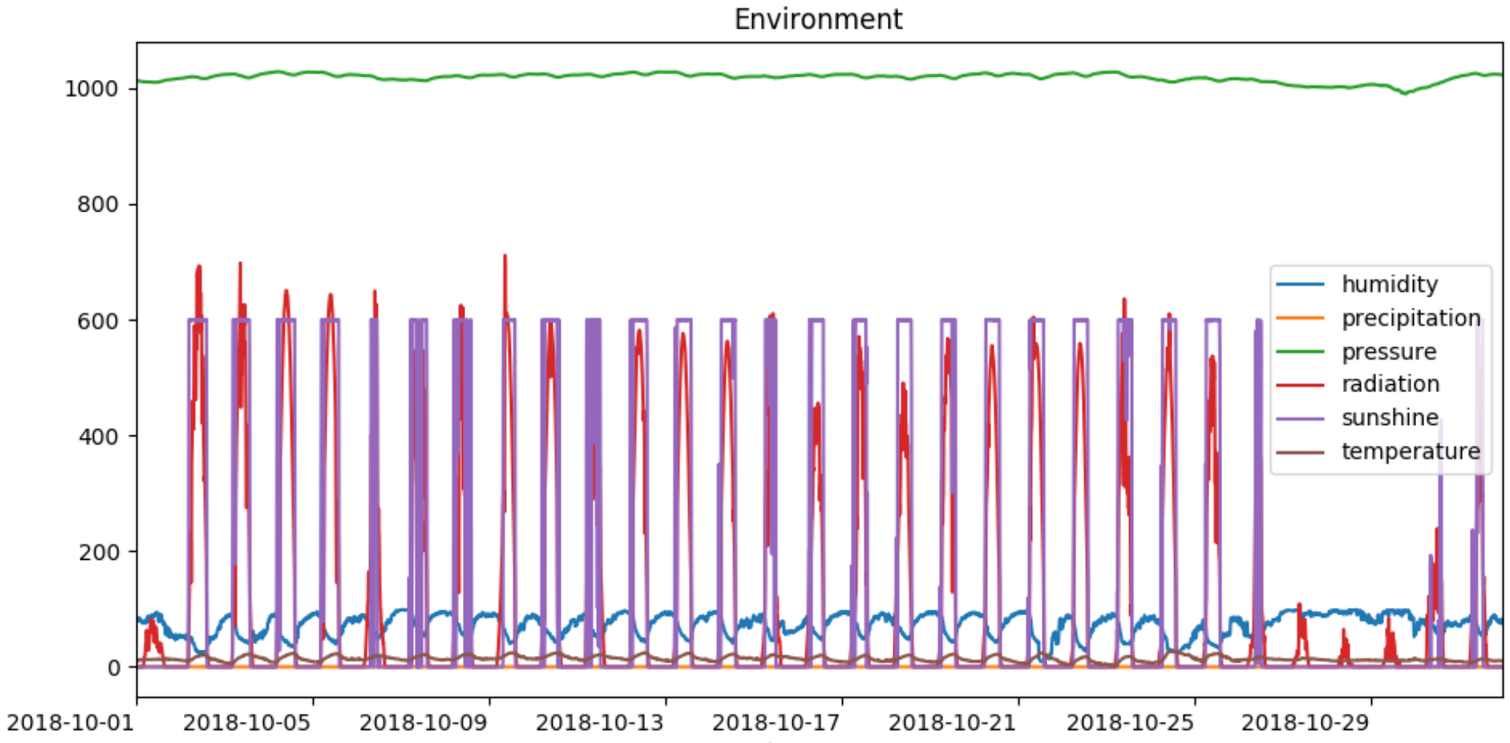
Perform EDA

ANALYZING IOT DATA IN PYTHON

Matthias Voppichler
IT Developer

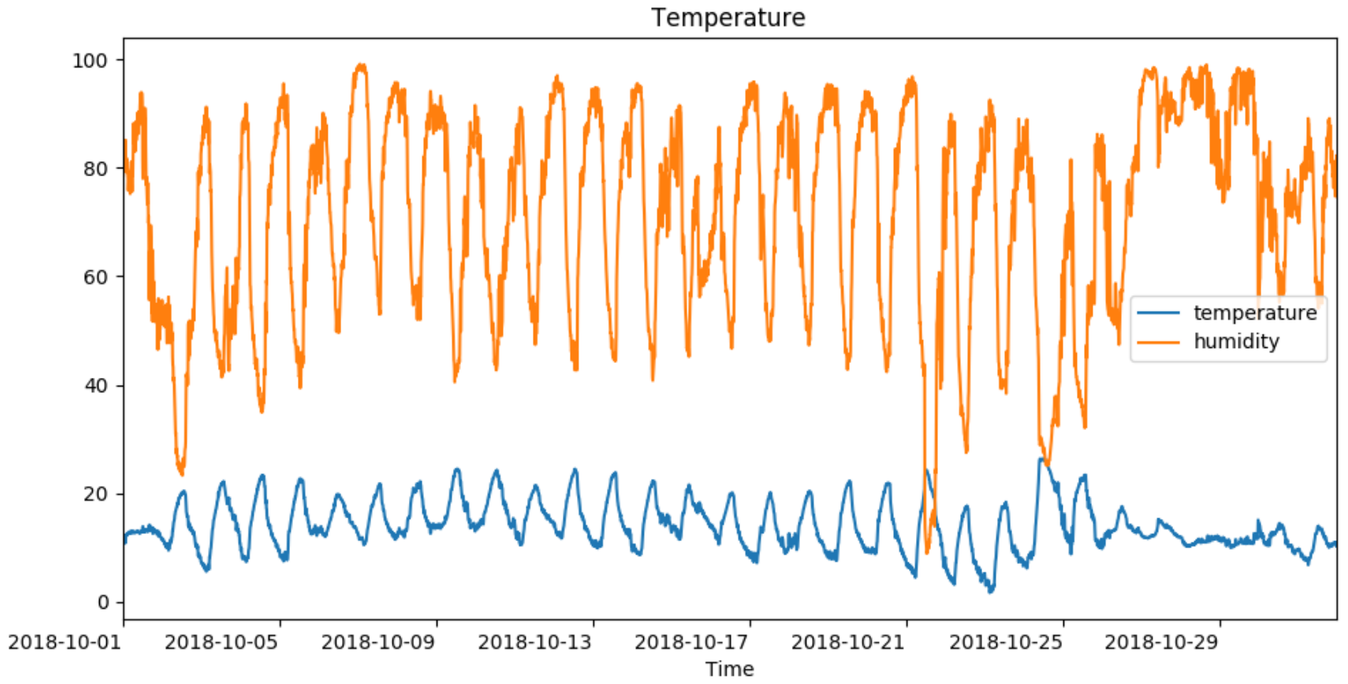
Plot dataframe

```
df.plot(title="Environment")
```



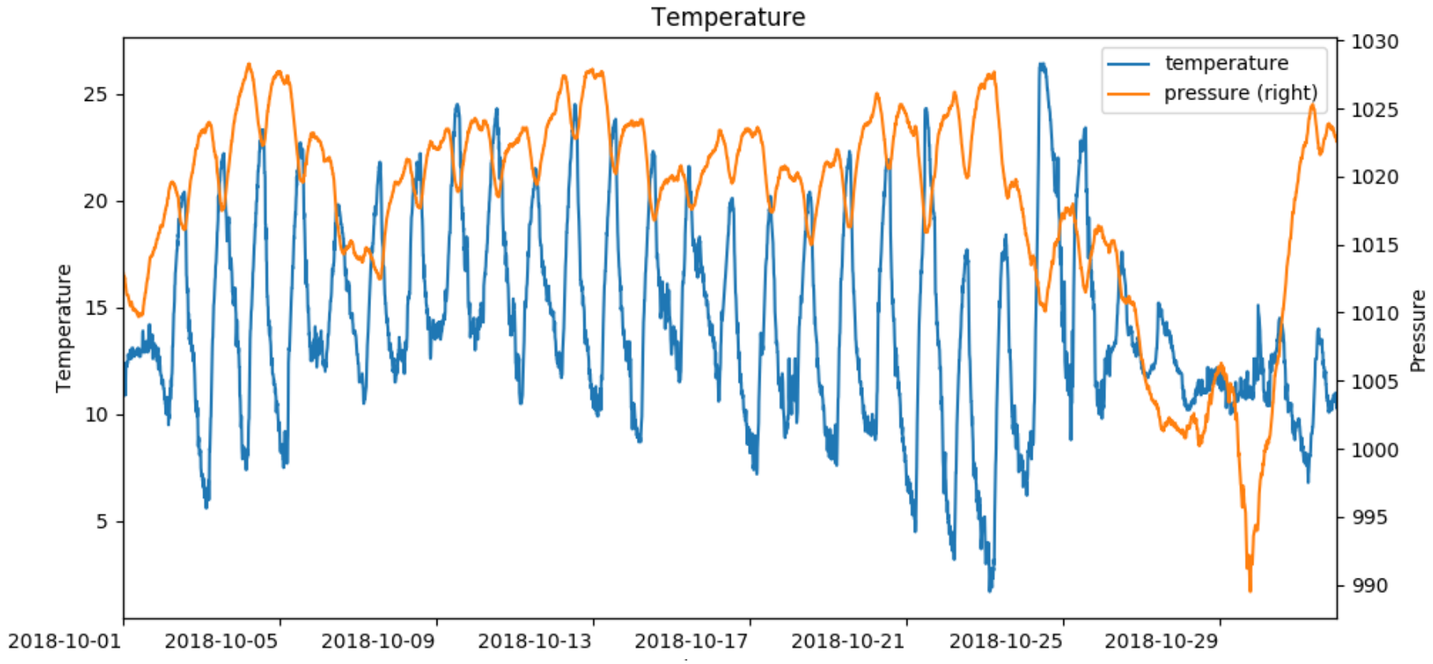
Line plot

```
df[["temperature", "humidity"]].plot(title="Environment")
plt.xlabel("Time")
```

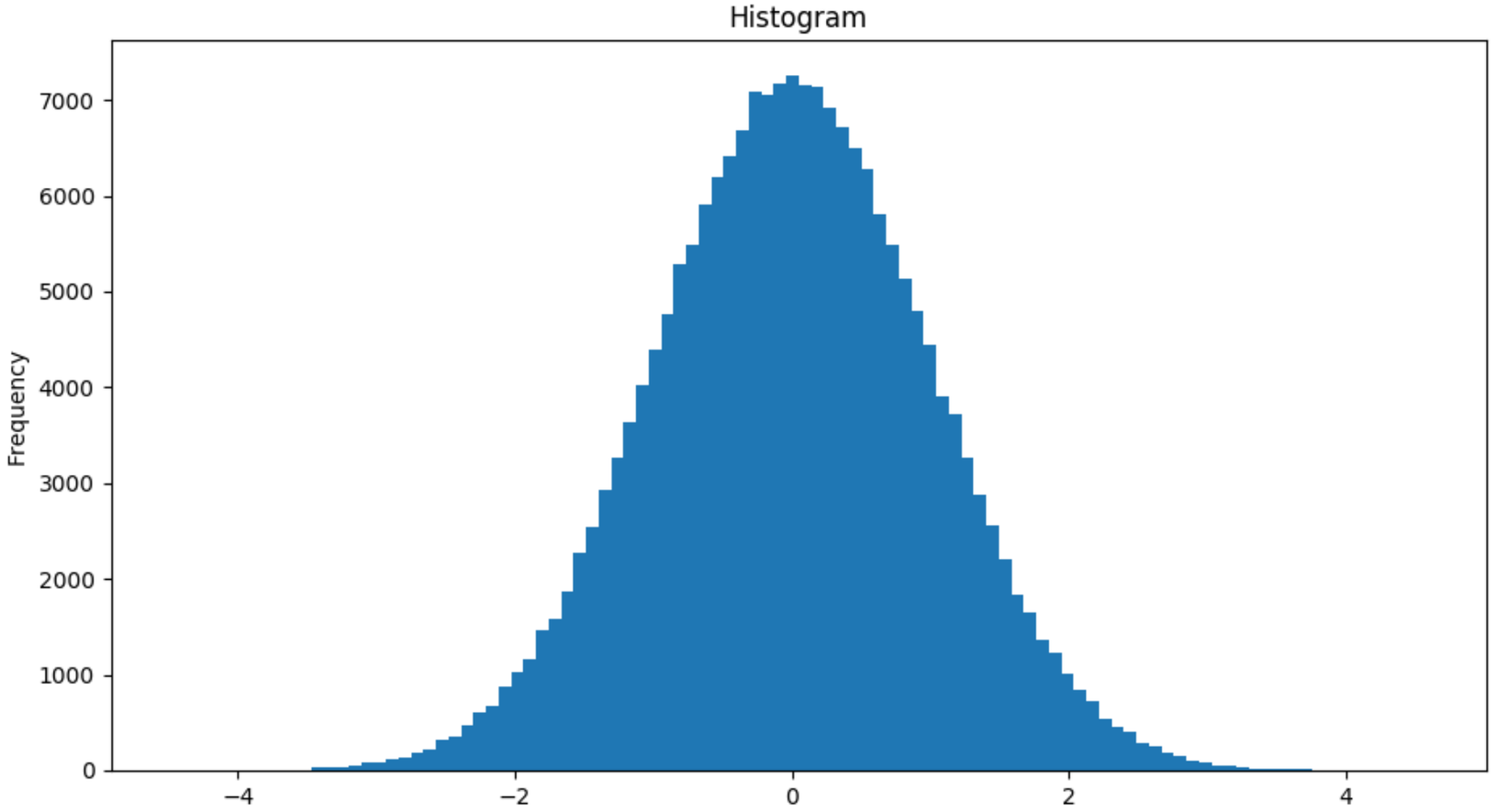


Secondary y

```
plt.ylabel('Temperature')
df[["temperature", "pressure"]].plot(title="Environment", secondary_y="pressure")
plt.ylabel('Pressure')
```

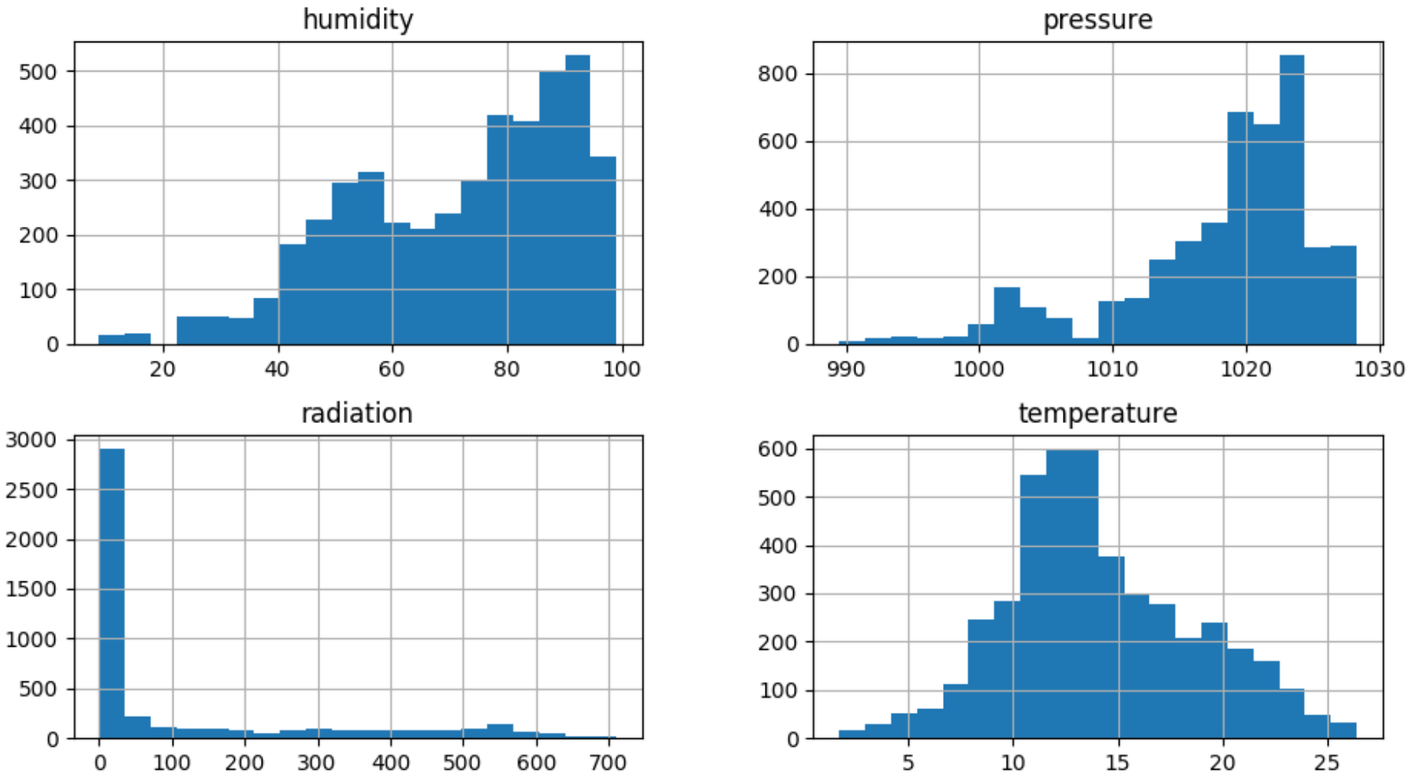


Histogram basics



Histogram

```
df.hist(bins=20)
```



Let's practice!

ANALYZING IOT DATA IN PYTHON

Clean Data

ANALYZING IOT DATA IN PYTHON

Matthias Voppichler
IT Developer

Missing data

Reasons for missing data from IoT devices

- Unstable network connection
- No power
- Other External factors

Times to deal with data quality

- During data collection
- During analysis

Dealing with missing data

Methods to deal with missing data

- fill
  - mean
  - median
  - forward-fill
  - backward-fill
- drop
- stop analysis

Detecting missing values

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 12 entries, 2018-10-15 08:00:00 to 2018-10-15 08:55:00
Data columns (total 3 columns):
temperature      8 non-null float64
humidity         8 non-null float64
precipitation    12 non-null float64
dtypes: float64(3)
memory usage: 384.0 bytes
```

Drop missing values

```
print(df.head())
```

```
                     temperature  humidity  precipitation
timestamp
2018-10-15 08:00:00         16.7      64.2            0.0
2018-10-15 08:05:00         16.6       NaN            0.0
2018-10-15 08:10:00         16.5      65.3            0.0
2018-10-15 08:15:00          NaN      65.0            0.0
2018-10-15 08:20:00         16.8      64.3            0.0
```

```
df.dropna()
```

```
                     temperature  humidity  precipitation
timestamp
2018-10-15 08:00:00         16.7      64.2            0.0
2018-10-15 08:10:00         16.5      65.3            0.0
2018-10-15 08:20:00         16.8      64.3            0.0
```

Fill missing values

```
df
```

```
                     temperature  humidity  precipitation
timestamp
2018-10-15 08:00:00         16.7      64.2            0.0
2018-10-15 08:05:00         16.6       NaN            0.0
2018-10-15 08:10:00         17.0      65.3            0.0
2018-10-15 08:15:00          NaN      65.0            0.0
2018-10-15 08:20:00         16.8      64.3            0.0
```

```
df.fillna(method="ffill")
```

```
                     temperature  humidity  precipitation
timestamp
2018-10-15 08:00:00         16.7      64.2            0.0
2018-10-15 08:05:00         16.6      64.2            0.0
2018-10-15 08:10:00         17.0      65.3            0.0
```

Interrupted Measurement

```
print(df.head())
```

```
timestamp            temperature  humidity
2018-10-15 00:00:00         13.5      84.7
2018-10-15 00:10:00         13.3      85.6
2018-10-15 00:20:00         12.9      88.8
2018-10-15 00:30:00         12.8      89.2
2018-10-15 00:40:00         13.0      87.7
```

```
print(df.isna().sum())
```

```
temperature    0
humidity       0
dtype: int64
```

```
df_res = df.resample("10min").last()
print(df_res.head())
```

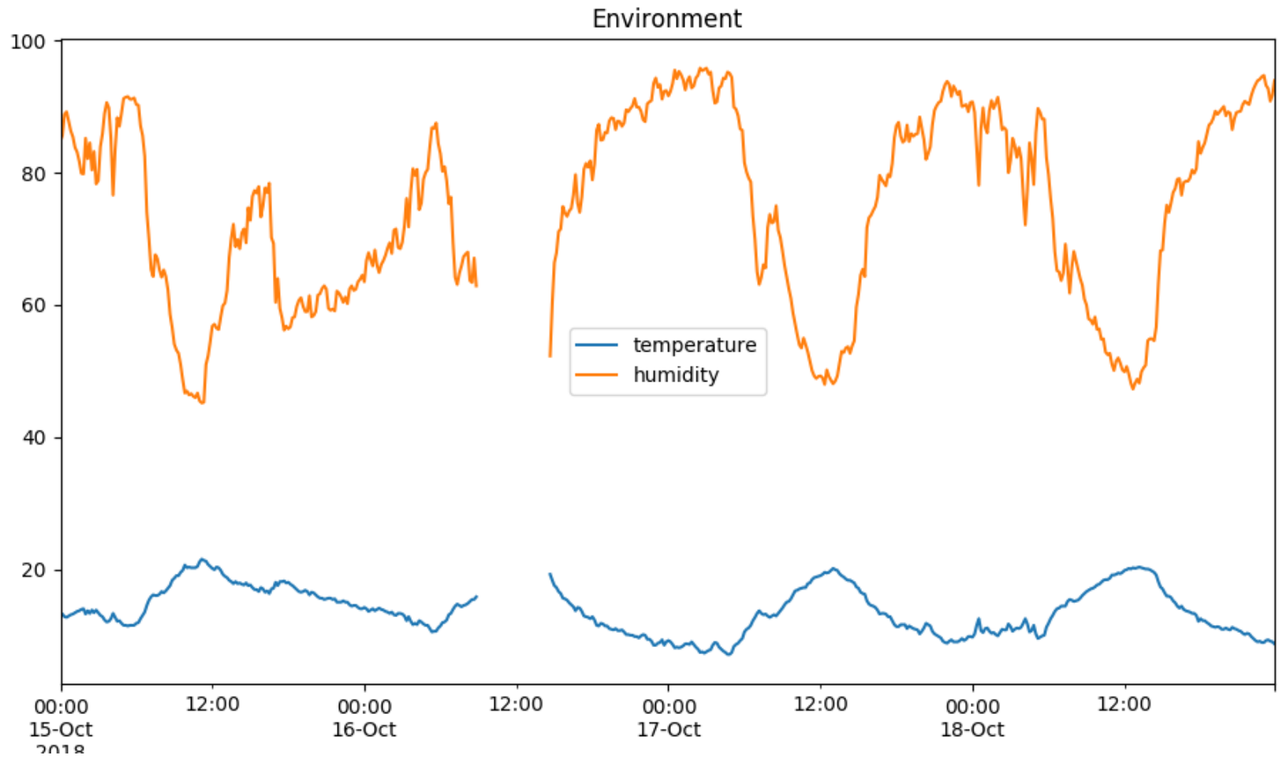
```
timestamp            temperature  humidity
2018-10-15 00:00:00         13.5      84.7
2018-10-15 00:10:00         13.3      85.6
2018-10-15 00:20:00         12.9      88.8
2018-10-15 00:30:00         12.8      89.2
2018-10-15 00:40:00         13.0      87.7
```

```
print(df_res.isna().sum())
```

```
temperature    34
humidity       34
dtype: int64
```

Interrupted Measurement

```
df_res.plot(title="Environment")
```



Let's practice!

ANALYZING IOT DATA IN PYTHON

Gather minimalistic incremental data

ANALYZING IOT DATA IN PYTHON

Matthias Voppichler
IT Developer

What is caching?

storing data

- After data stream collection
- Observation by observation
  - Creates high load on Disks
- **Use caching**

Caching

```
cache = []

def on_message(client, userdata, message):
    data = json.loads(message.payload)
    cache.append(data)

    if len(cache) > MAX_CACHE:

        with Path("data.txt").open("a") as f:
            f.writelines(cache)
        cache.clear()

# Connect function to mqtt datastream
subscribe.callback(on_message,
                   topics="datacamp/energy",
                   hostname=MQTT_HOST)
```

Simplistic datastreams

```
C331,6020
M640,104
C331,6129
M640,180
C331,6205
M640,256
```

Observation Timestamp

- "timestamp in payload"
- `message.timestamp`
- `datetime.now()`

Observation Timestamp

```
def on_message(client, userdata, message):
    publishtime = message.timestamp

    consume_time = datetime.utcnow()
```

pd.to_datetime()

```
print(df.head())
```

```
   timestamp          device  val
0  1540535443083        C331  347069.305500
1  1540535460858        C331  347069.381205
```

```
import pandas as pd
df["timestamp"] = pd.to_datetime(df["timestamp"], unit="ms")
```

```
  timestamp                  device  val
0 2018-10-26 06:30:43.083      C331  347069.305500
1 2018-10-26 06:31:00.858      C331  347069.381205
```

Let's practice!

ANALYZING IOT DATA IN PYTHON

Prepare and visualize incremental data

ANALYZING IOT DATA IN PYTHON

Matthias Voppichler
IT Developer

Data preparation

- Pivot data
- Resample
- Apply `diff()`
- Apply `pct_change()`

Data structure

```
print(data.head())
```

```
                 timestamp device   value
0 2018-10-26 06:30:42.817   C331  6020.0
1 2018-10-26 06:30:43.083   M640   104.0
2 2018-10-26 06:31:00.858   M640   126.0
3 2018-10-26 06:31:10.254   C331  6068.0
4 2018-10-26 06:31:10.474   M640   136.0
```

Pivot table

```
pd.pivot_table(df, index="timestamp",
                   columns="device",
                   values="value")
```

timestamp	device	value
2018-10-26	M640	10
2018-10-26	C331	11
2018-10-27	C331	13
2018-10-27	M640	12

timestamp	C331	M640
2018-10-26	11	10
2018-10-27	13	12

Apply pivot table

```
                 timestamp device   value
0 2018-10-26 06:30:42.817   C331  6020.0
1 2018-10-26 06:30:43.083   M640   104.0
2 2018-10-26 06:31:00.858   M640   126.0
3 2018-10-26 06:31:10.254   C331  6068.0
4 2018-10-26 06:31:10.474   M640   136.0
```

```
data = pd.pivot_table(data, columns="device", values="value", index="timestamp")
print(data.head()
```

```
device                    C331   M640
timestamp
2018-10-26 06:30:42.817  6020.0    NaN
2018-10-26 06:30:43.083     NaN  104.0
2018-10-26 06:31:00.858     NaN  126.0
2018-10-26 06:31:10.254  6068.0    NaN
2018-10-26 06:31:10.474     NaN  136.0
```

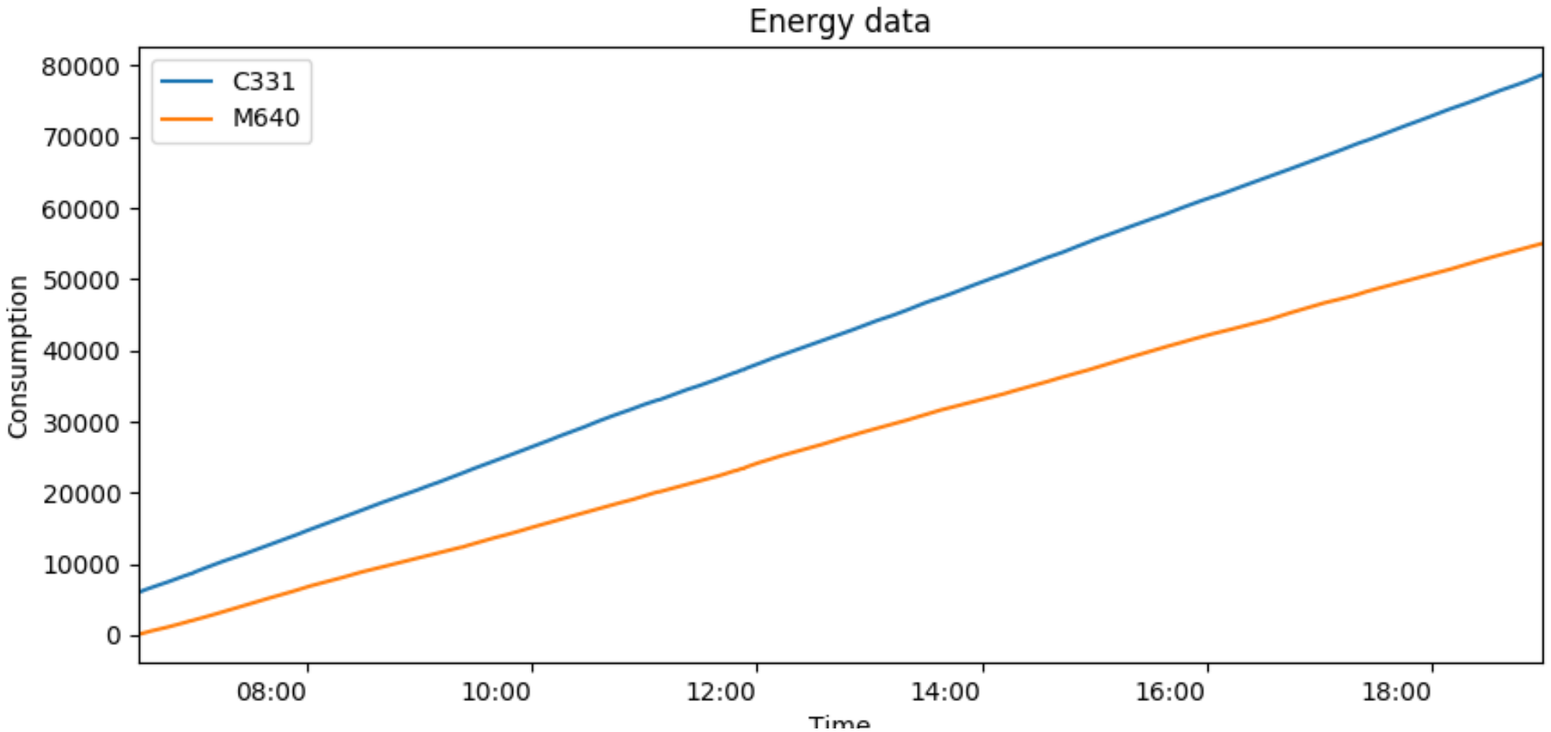
Resample

```
# Resample dataframe to 1min
df = data.resample("1min").max().dropna()
print(df.head())
```

```
device                C331   M640
timestamp
2018-10-26 06:30:00  6020.0  104.0
2018-10-26 06:31:00  6129.0  180.0
2018-10-26 06:32:00  6205.0  256.0
2018-10-26 06:33:00  6336.0  332.0
2018-10-26 06:34:00  6431.0  402.0
```

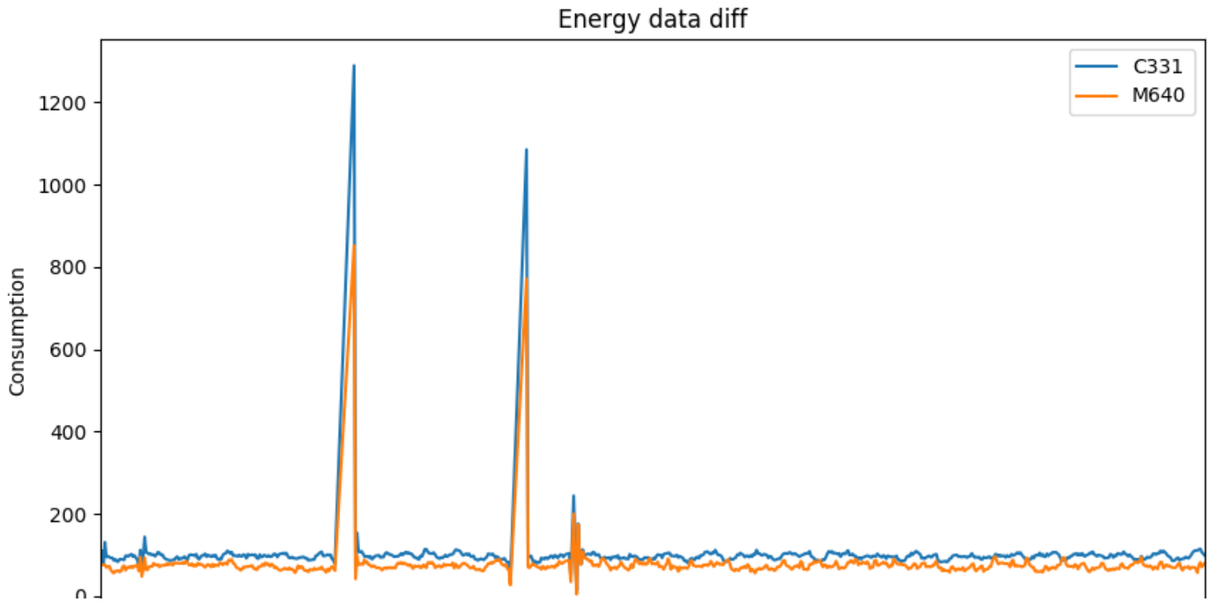
Visualize data

```
data.plot()
plt.show()
```



pd.diff()

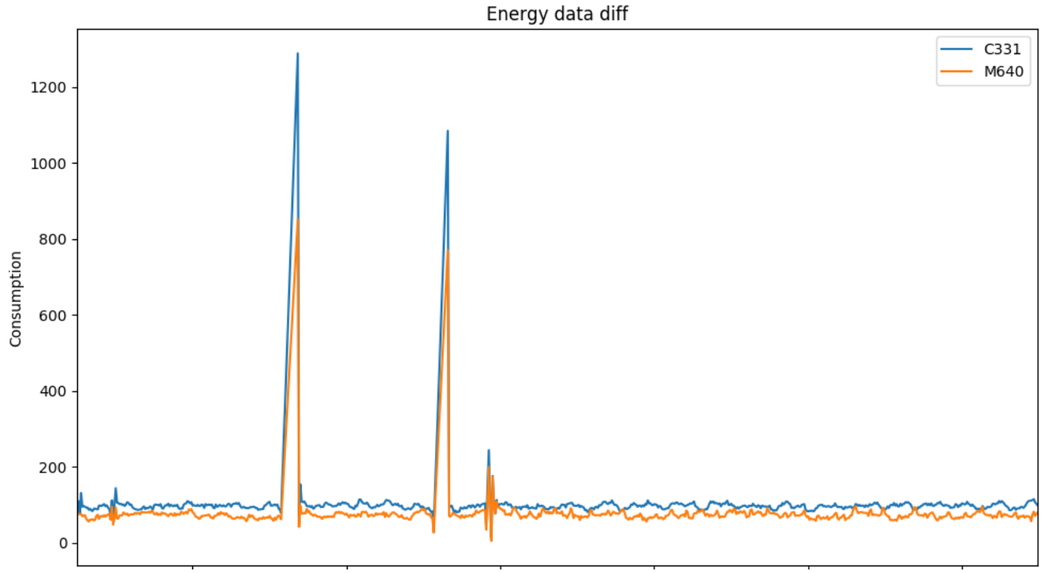
```
# Difference
df_diff = data.diff(1)
df_diff.plot()
plt.show()
```

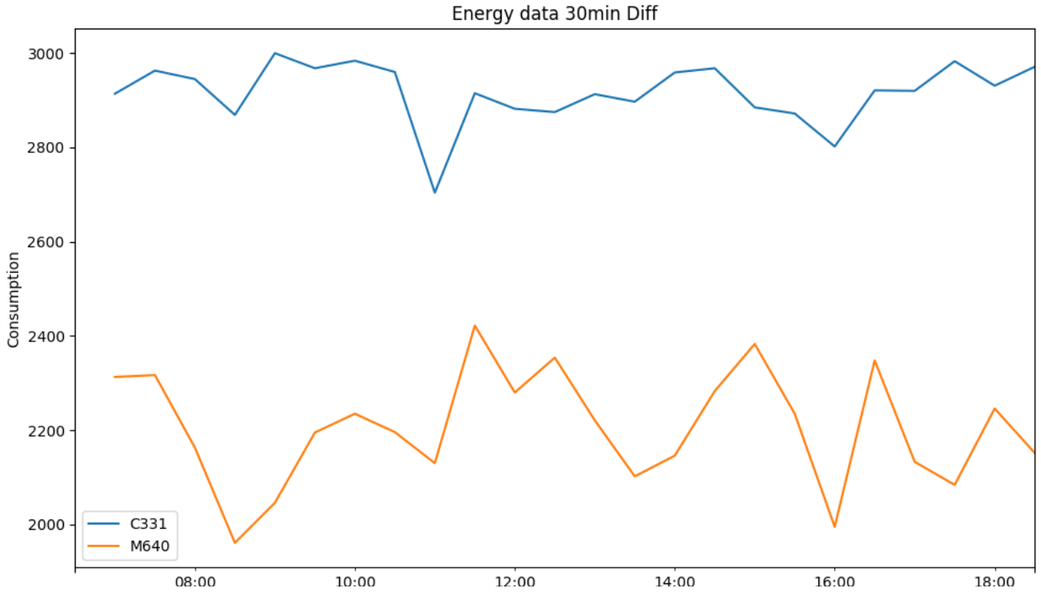


Data analysis - difference

```
# Difference
df_diff = data.diff()
df_diff.plot()
plt.show()
```

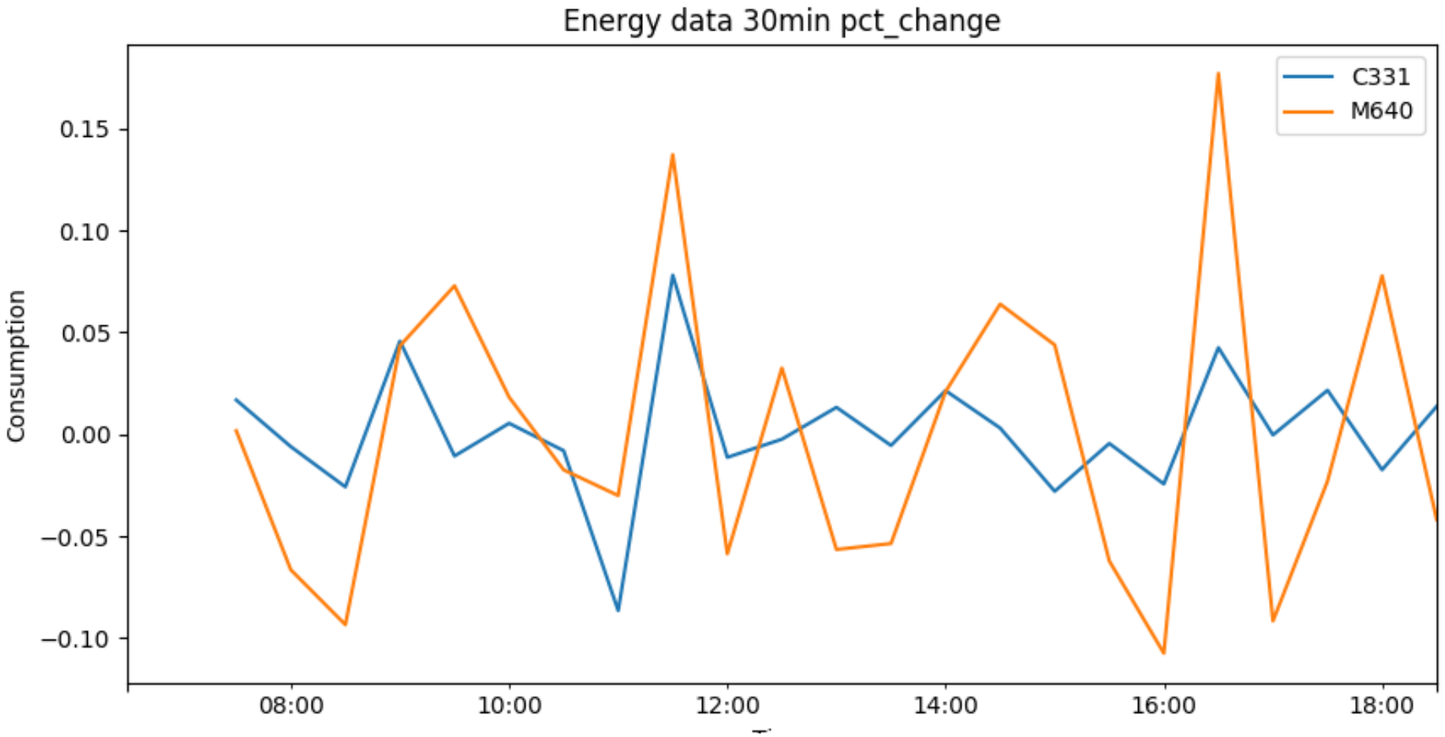
```
# Resampled difference
df = data.resample('30min').max()
df_diff = df.diff()
df_diff.plot()
plt.show()
```





Change percentage

```
df_pct = df_diff.pct_change()
df_pct.plot()
```



Let's Practice

ANALYZING IOT DATA IN PYTHON