**MESUREMENT ENERGY CONSUMPTION:**

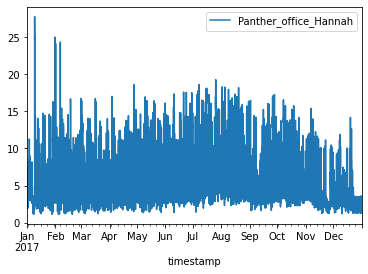
**PHASE-4:**

**Load data:**

office\_example\_prediction\_data.plot()

<AxesSubplot:xlabel='timestamp'>

Output:



weather\_data = pd.read\_csv("../input/buildingdatagenomeproject2/weather.csv", index\_col='timestamp', parse\_dates=True)

weather\_data\_site = weather\_data[weather\_data.site\_id == 'Panther'].truncate(before='2017-01-01')

weather\_data\_site.info()

Output:

<class 'pandas.core.frame.DataFrame'>

DatetimeIndex: 8760 entries, 2017-01-01 00:00:00 to 2017-12-31 23:00:00

Data columns (total 9 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 site\_id 8760 non-null object

1 airTemperature 8760 non-null float64

2 cloudCoverage 5047 non-null float64

3 dewTemperature 8760 non-null float64

4 precipDepth1HR 8752 non-null float64

5 precipDepth6HR 329 non-null float64

6 seaLvlPressure 8522 non-null float64

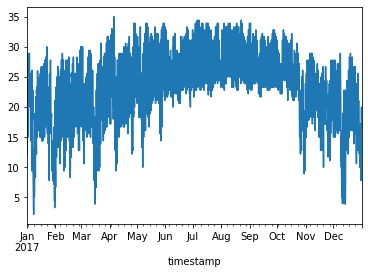
7 windDirection 8511 non-null float64

8 windSpeed 8760 non-null float64

weather\_hourly = weather\_data\_site.resample("H").mean()weather\_hourly\_nooutlier = weather\_hourly[weather\_hourly > -40]weather\_hourly\_nooutlier\_nogaps = weather\_hourly\_nooutlier.fillna(method='ffill')

temperature = weather\_hourly\_nooutlier\_nogaps["airTemperature"]

temperature.plot()



## Create Train and Test Datasets

The model is given a set of data that will be used to ****train**** the model to predict a specific objectice. In this case, we will use a few simple time series features as well as outdoor air temperature to predict how much energy a building uses.

For this demonstration, we will use three months of data from April, May, and June to prediction July.

training\_months = [4,5,6]test\_months = [7]

trainingdata = office\_example\_prediction\_data[office\_example\_prediction\_data.index.month.isin(training\_months)]testdata = office\_example\_prediction\_data[office\_example\_prediction\_data.index.month.isin(test\_months)]

trainingdata.info()

**Output:**

<class 'pandas.core.frame.DataFrame'>

DatetimeIndex: 2184 entries, 2017-04-01 00:00:00 to 2017-06-30 23:00:00

Data columns (total 1 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Panther\_office\_Hannah 2184 non-null float64

dtypes: float64(1)

memory usage: 34.1 KB

## Encoding Categorical Variables:

train\_features = pd.concat([pd.get\_dummies(trainingdata.index.hour),

pd.get\_dummies(trainingdata.index.dayofweek),

pd.DataFrame(temperature[temperature.index.month.isin(training\_months)].values)], axis=1).dropna()

train\_features.head()

Output:

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ... | 22 | 23 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 0 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 21.7 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 21.0 |
| 2 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 18.9 |
| 3 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 20.6 |
| 4 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 21.0 |

## Train a K-Neighbor Model

This model was chosen after following the process in the cheat sheet until a model that worked and provided good results was found.

test\_features = np.array(pd.concat([pd.get\_dummies(testdata.index.hour),

pd.get\_dummies(testdata.index.dayofweek),

pd.DataFrame(temperature[temperature.index.month.isin(test\_months)].values)], axis=1).dropna())

**TEAM:**

**1.M.MANIKANDAN**

**2.R.MANIKANDAN**

**3.P.ANTONY FRANCIS**

**4.R.CHERMA RAJA**

**5.B.PREM KUMAR**