In [1]:
```python
# import library
import numpy as np
import pandas as pd
import os
import matplotlib
import matplotlib.pyplot as plt
%matplotlib inline
```

Natural ventilation cooling potential

In [2]:
```python
def getPandasFromFile(path, fileName, theSkipRow):
    path = os.path.join(path , fileName)
    return pd.read_csv(path, skiprows= theSkipRow , header=None)
def SavePandasToCSV(d, path):
    d.to_csv(path)
    return "done!!"
def getPandasFromFile_my(path, fileName):
    path = os.path.join(path , fileName)
    return path
```

In [3]:
```python
d = getPandasFromFile("",'USA_MA_Boston-Logan.Intl.AP.725090_TMY3.epw', 8)
theDf =d.rename(index=str, columns={0:"year", 1: "Month", 2: "Day", 3:"Hour",4:"Minute",6:"D
B_temp",
                                    7:"Dew_Point",
                                    8:"RH",9:"P", 10:"Horiz_Rad",11:"Normal_Rad",12:"Sky_Ra
d",
                                    13:"G_Horiz_Rad",14:"Dir_Normal_Rad",15:"Diff_Horiz_Ra
d",
                                    16:"G_Horiz_Illu",17:"Dir_Normal_Illu",18:"Diff_Horiz_I
llu",
                                    19:"Zenith_Illu",20:"Wind_Direction",21:"Wind_Speed",
                                    22:"Total_Sky_Cov",23:"Opaque_Sky_Cov",24:"Visibility",
                                    25:"field_Ceiling_H",26:"Whtr_Observ",27:"Whtr_Codes",
                                    28:"Pred_Water",29:"AeroesOptical_D",30:"Snow_Depth",
                                    31:"Days_Since_Snow"
                                   })
```

In [5]:
```python
#theDf.head()
```

Out[5]:

	year	Month	Day	Hour	Minute	5	DB_temp	Dew_Point	RH	P	...	field_Ceiling_H	Whtr_Observ	Whtr_Cod
0	1976	1	1	1	0	?9?9?9? 9E0?9? 9?9?9? 9?9?9? 9?9?9? 9?9?9? 9? 9*9*9*9...	1.7	-3.9	67	100900	...	3960	9	9999999
1	1976	1	1	2	0	?9?9?9? 9E0?9? 9?9?9? 9?9?9? 9?9?9? 9?9?9? 9? 9*9*9*9...	1.7	-3.9	67	100800	...	3960	9	9999999
2	1976	1	1	3	0	?9?9?9? 9E0?9? 9?9?9? 9?9?9? 9?9?9? 9?9?9? 9? 9*9*9*9...	1.1	-3.9	70	100800	...	3960	9	9999999
3	1976	1	1	4	0	?9?9?9? 9E0?9? 9?9?9? 9?9?9? 9?9?9? 9?9?9? 9? 9*9*9*9...	1.1	-3.9	70	100700	...	3960	9	9999999
4	1976	1	1	5	0	?9?9?9? 9E0?9? 9?9?9? 9?9?9? 9?9?9? 9?9?9? 9? 9*9*9*9...	1.1	-4.4	67	100700	...	3960	9	9999999

5 rows × 35 columns

In [4]:
```python
NewDf = theDf[['DB_temp',"Dew_Point", 'RH','Wind_Direction', 'Wind_Speed'
              ]]
NewDf.head()
```

Out[4]:

	DB_temp	Dew_Point	RH	Wind_Direction	Wind_Speed
0	1.7	-3.9	67	360	6.2
1	1.7	-3.9	67	350	5.7
2	1.1	-3.9	70	360	6.2
3	1.1	-3.9	70	10	5.7
4	1.1	-4.4	67	10	5.7

In [6]:
```python
#WSLst = NewDf["Wind_Speed"].tolist()
#Sum = 0
#for i in range(0,len(WSLst)):
#    Sum += float(WSLst[i])
#print("my Local Wind Speed is: " , Sum/len(WSLst))
```

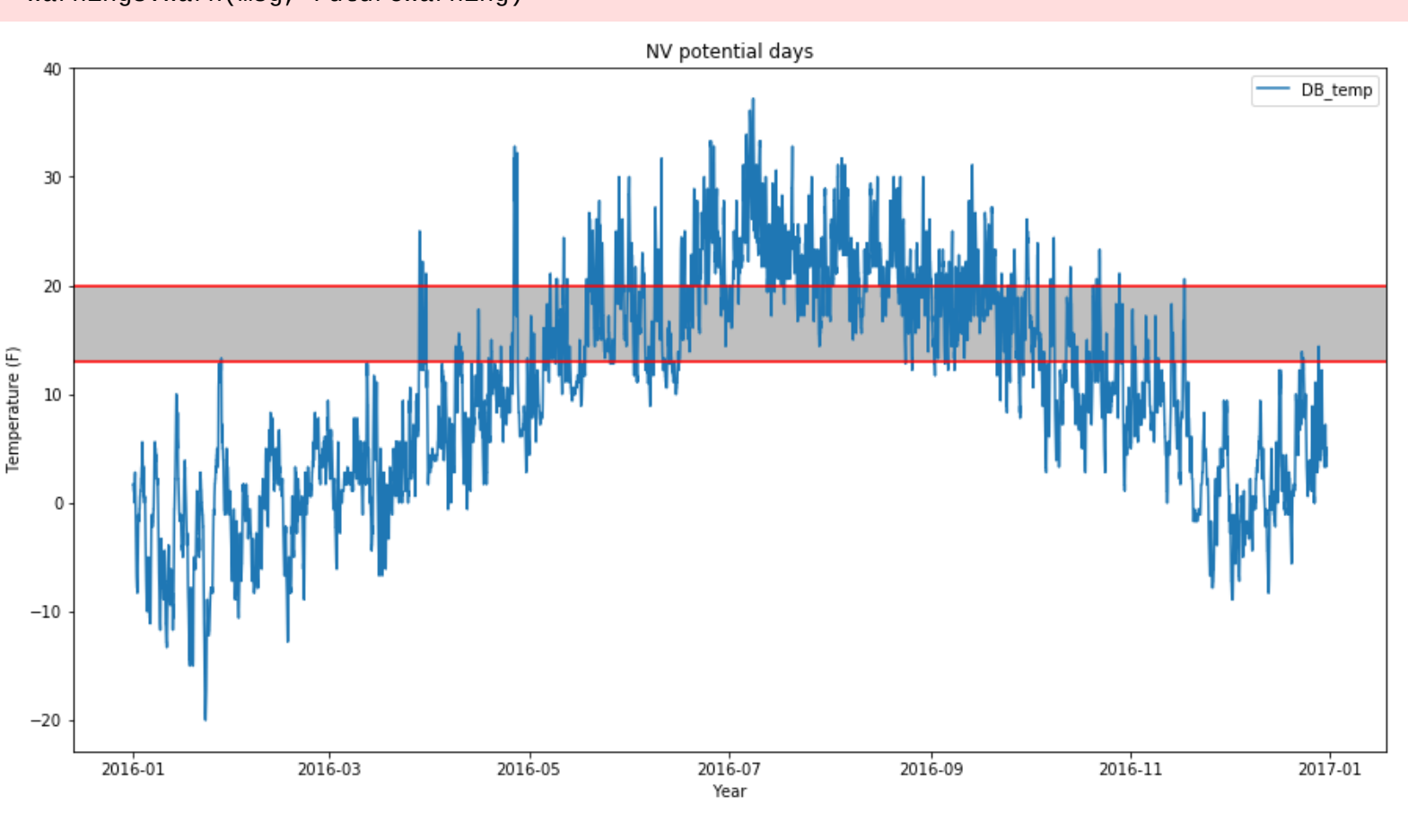
```
my Local Wind Speed is:  5.484703196346415
```

In [11]:
```python
#WSLst = NewDf["Wind_Speed"].tolist()
#Sum = 0
#Count = 0
#for i in range(0,len(WSLst)):
#    if(NewDf["DB_temp"][i]>1.3 and NewDf["DB_temp"][i]<2.0):
#        Sum += float(WSLst[i])
#        Count +=1
#
#print("my Local Wind Speed is: " , Sum/Count) #for further input
```

```
my Local Wind Speed is:  5.386697247706431
```

In [12]:
```python
x = NewDf.index
temp = NewDf["DB_temp"]

plt.figure(figsize=(15,8))
myTime = pd.date_range('1/1/2016', periods=8760, freq='H')

plt.plot(myTime,temp)
plt.title('NV potential days')
plt.xlabel('Year')
plt.ylabel('Temperature (F)')
plt.legend(loc='best')

plt.axhline(13, color='r', label=r'True $\beta_1$')
plt.axhline(20, color='r', label=r'True $\beta_1$')
plt.axhspan(13, 20, facecolor='0.5', alpha=0.5,label=r'True $\beta_1$')

plt.show()
```

```
E:\soft\Anaconda3\lib\site-packages\pandas\plotting\_converter.py:129: FutureWarning: Using a
n implicitly registered datetime converter for a matplotlib plotting method. The converter wa
s registered by pandas on import. Future versions of pandas will require you to explicitly re
gister matplotlib converters.

To register the converters:
        >>> from pandas.plotting import register_matplotlib_converters
        >>> register_matplotlib_converters()
  warnings.warn(msg, FutureWarning)
```



In [13]:
```python
energyDF = pd.read_csv('Energy_Boston.csv', sep=",", header=0)
```

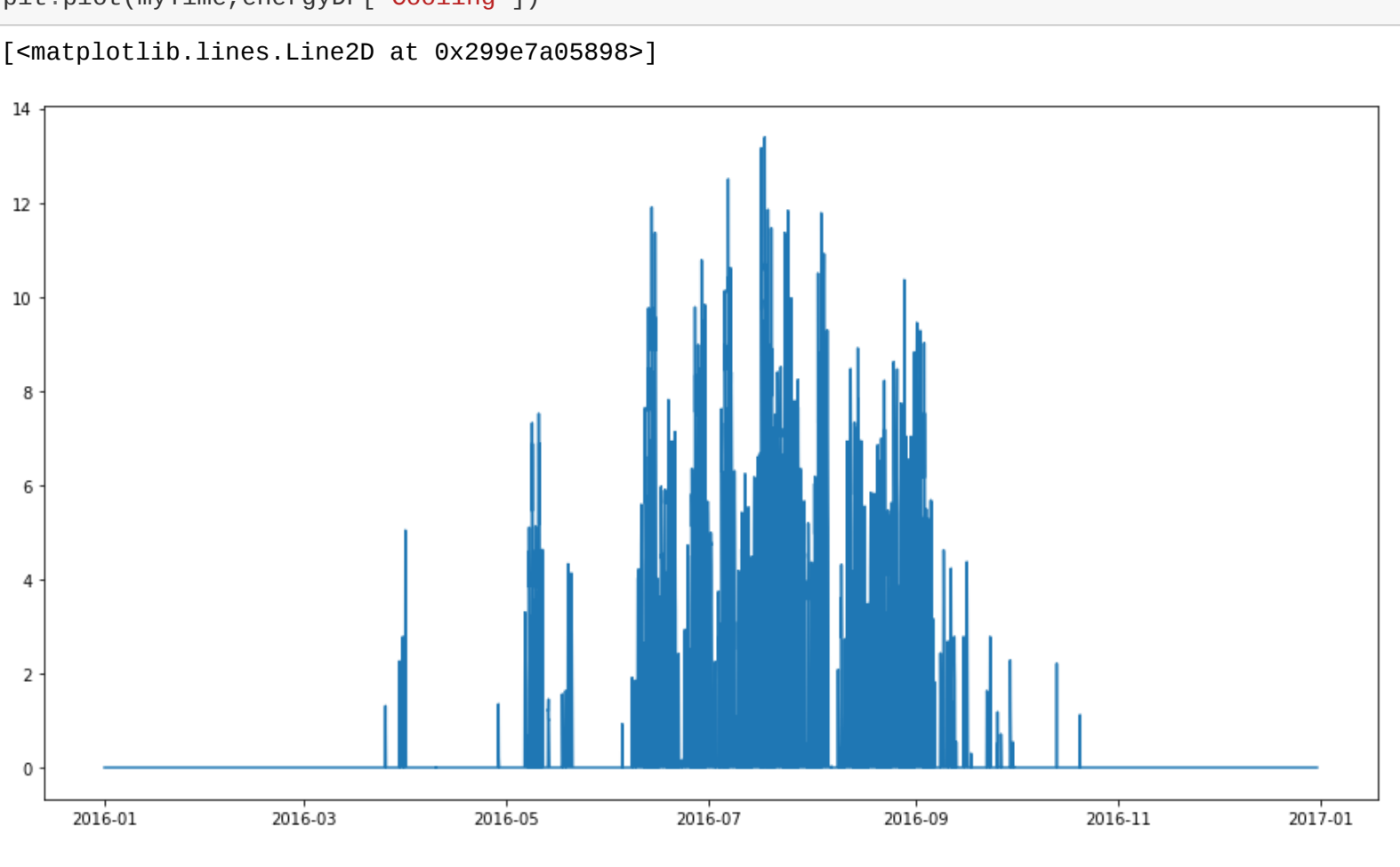
In [14]:
```python
energyDF.tail()
```

Out[14]:

	Date/Time	Cooling	Heating
8755	12/31 20:00:00	0.0	6.70
8756	12/31 21:00:00	0.0	6.97
8757	12/31 22:00:00	0.0	0.00
8758	12/31 23:00:00	0.0	0.98
8759	12/31 24:00:00	0.0	2.15

In [15]:
```python
plt.figure(figsize=(15,8))
plt.plot(myTime,energyDF["Cooling"])
```

Out[15]: [<matplotlib.lines.Line2D at 0x299e7a05898>]



In [17]:
```python
## your code here
#
#tempLst = temp.tolist()
#NVLst = []
#for i in range(0,len(tempLst)):
#    if(tempLst[i]>13 and tempLst[i]<23):
#        NVLst.append(0)
#    else:
#        NVLst.append(1)
#
#np.unique(NVLst)
##NVLst
```

Out[17]: array([0, 1])

In [18]:
```python
## your code here

#energyDF["Cooling_NV"] = energyDF["Cooling"]*NVLst
#energyDF
```

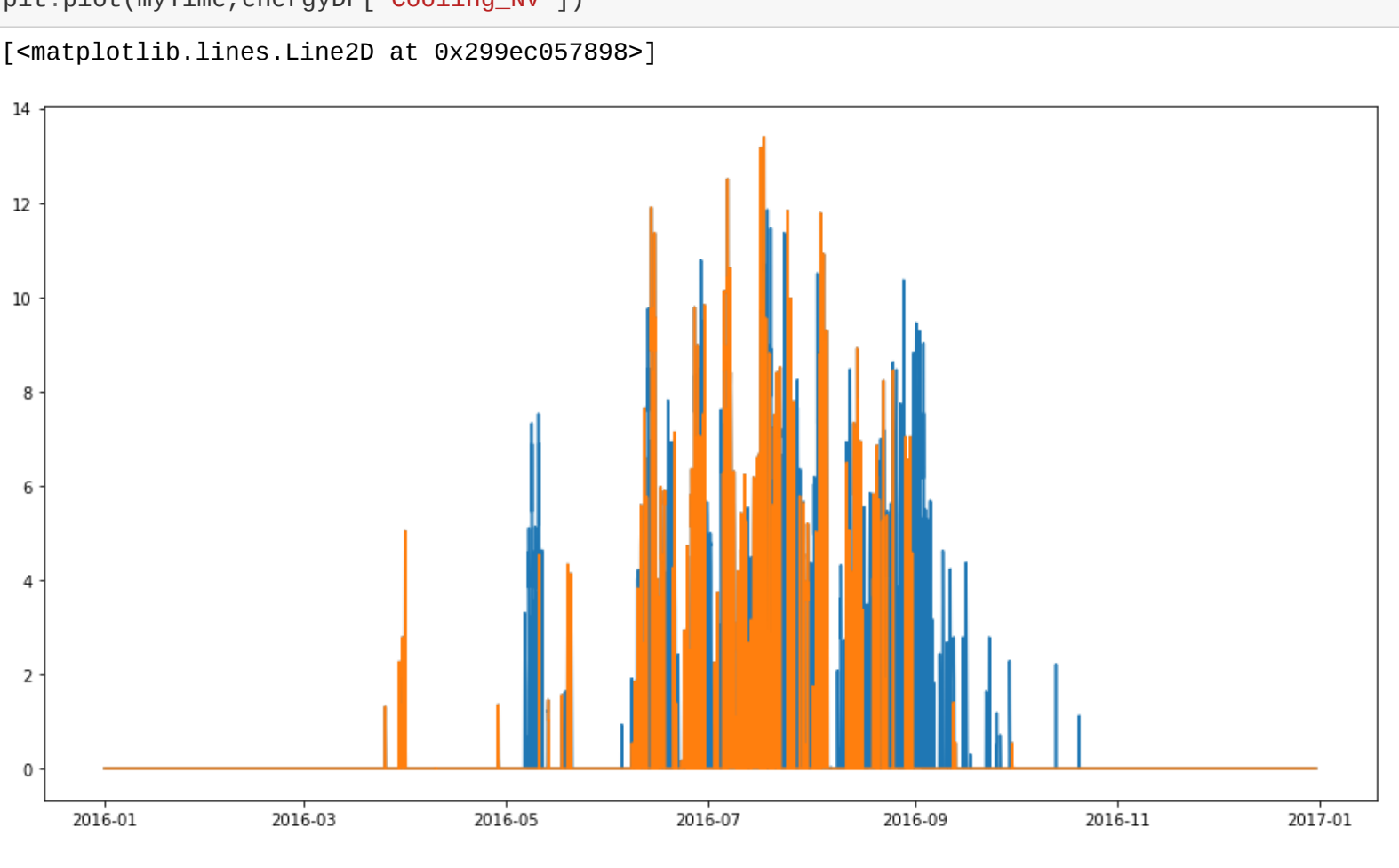
Out[18]:

	Date/Time	Cooling	Heating	Cooling_NV
0	01/01 01:00:00	0.0	3.28	0.0
1	01/01 02:00:00	0.0	3.98	0.0
2	01/01 03:00:00	0.0	4.41	0.0
3	01/01 04:00:00	0.0	4.63	0.0
4	01/01 05:00:00	0.0	4.75	0.0
5	01/01 06:00:00	0.0	4.87	0.0
6	01/01 07:00:00	0.0	5.04	0.0
7	01/01 08:00:00	0.0	18.96	0.0
8	01/01 09:00:00	0.0	15.41	0.0
9	01/01 10:00:00	0.0	11.48	0.0
10	01/01 11:00:00	0.0	10.01	0.0
11	01/01 12:00:00	0.0	8.89	0.0
12	01/01 13:00:00	0.0	8.41	0.0
13	01/01 14:00:00	0.0	7.91	0.0
14	01/01 15:00:00	0.0	7.66	0.0
15	01/01 16:00:00	0.0	7.81	0.0
16	01/01 17:00:00	0.0	8.38	0.0
17	01/01 18:00:00	0.0	8.75	0.0
18	01/01 19:00:00	0.0	8.70	0.0
19	01/01 20:00:00	0.0	9.16	0.0
20	01/01 21:00:00	0.0	8.99	0.0
21	01/01 22:00:00	0.0	0.35	0.0
22	01/01 23:00:00	0.0	2.03	0.0
23	01/01 24:00:00	0.0	2.91	0.0
24	01/02 01:00:00	0.0	3.34	0.0
25	01/02 02:00:00	0.0	3.64	0.0
26	01/02 03:00:00	0.0	3.83	0.0
27	01/02 04:00:00	0.0	4.13	0.0
28	01/02 05:00:00	0.0	4.24	0.0
29	01/02 06:00:00	0.0	17.83	0.0
...	...	...	...	...
8730	12/30 19:00:00	0.0	6.85	0.0
8731	12/30 20:00:00	0.0	6.84	0.0
8732	12/30 21:00:00	0.0	7.03	0.0
8733	12/30 22:00:00	0.0	0.00	0.0
8734	12/30 23:00:00	0.0	0.68	0.0
8735	12/30 24:00:00	0.0	1.65	0.0
8736	12/31 01:00:00	0.0	2.44	0.0
8737	12/31 02:00:00	0.0	2.62	0.0
8738	12/31 03:00:00	0.0	3.01	0.0
8739	12/31 04:00:00	0.0	3.38	0.0
8740	12/31 05:00:00	0.0	3.98	0.0
8741	12/31 06:00:00	0.0	4.23	0.0
8742	12/31 07:00:00	0.0	4.55	0.0
8743	12/31 08:00:00	0.0	18.32	0.0
8744	12/31 09:00:00	0.0	13.44	0.0
8745	12/31 10:00:00	0.0	7.46	0.0
8746	12/31 11:00:00	0.0	4.59	0.0
8747	12/31 12:00:00	0.0	2.44	0.0
8748	12/31 13:00:00	0.0	2.32	0.0
8749	12/31 14:00:00	0.0	1.68	0.0
8750	12/31 15:00:00	0.0	1.76	0.0
8751	12/31 16:00:00	0.0	3.02	0.0
8752	12/31 17:00:00	0.0	4.66	0.0
8753	12/31 18:00:00	0.0	5.76	0.0
8754	12/31 19:00:00	0.0	6.22	0.0
8755	12/31 20:00:00	0.0	6.70	0.0
8756	12/31 21:00:00	0.0	6.97	0.0
8757	12/31 22:00:00	0.0	0.00	0.0
8758	12/31 23:00:00	0.0	0.98	0.0
8759	12/31 24:00:00	0.0	2.15	0.0

8760 rows × 4 columns

In [37]:
```python
plt.figure(figsize=(15,8))
plt.plot(myTime,energyDF["Cooling"])
plt.plot(myTime,energyDF["Cooling_NV"])
```

Out[37]: [<matplotlib.lines.Line2D at 0x299ec057898>]



In [22]:
```python
energyDF.describe() #the mean cooling decreases from 0.57 to 0.26
```

Out[22]:

	Cooling	Heating	Cooling_NV
count	8760.000000	8760.000000	8760.000000
mean	0.571703	3.368909	0.261039
std	1.769531	5.759228	1.281257
min	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000
50%	0.000000	0.000000	0.000000
75%	0.000000	4.920000	0.000000
max	13.390000	36.150000	13.390000

Imputing missing values

In [23]:
```python
df = pd.read_csv('HouseZero_full.csv', sep=",", header=0) # missing dataset
df= df[:4000]
refer = pd.read_csv('haravrd_hourly.csv', sep=",", header=0) #reference dataset
refer= refer[:4000]
```

In [24]:
```python
df.head()
```

Out[24]:

	Temp	WS	WD
0	4.400000	2.600000	178.000000
1	4.400000	2.283333	182.416667
2	6.311000	2.106667	175.783333
3	7.026988	2.037349	185.168675
4	7.938321	1.109489	235.072993

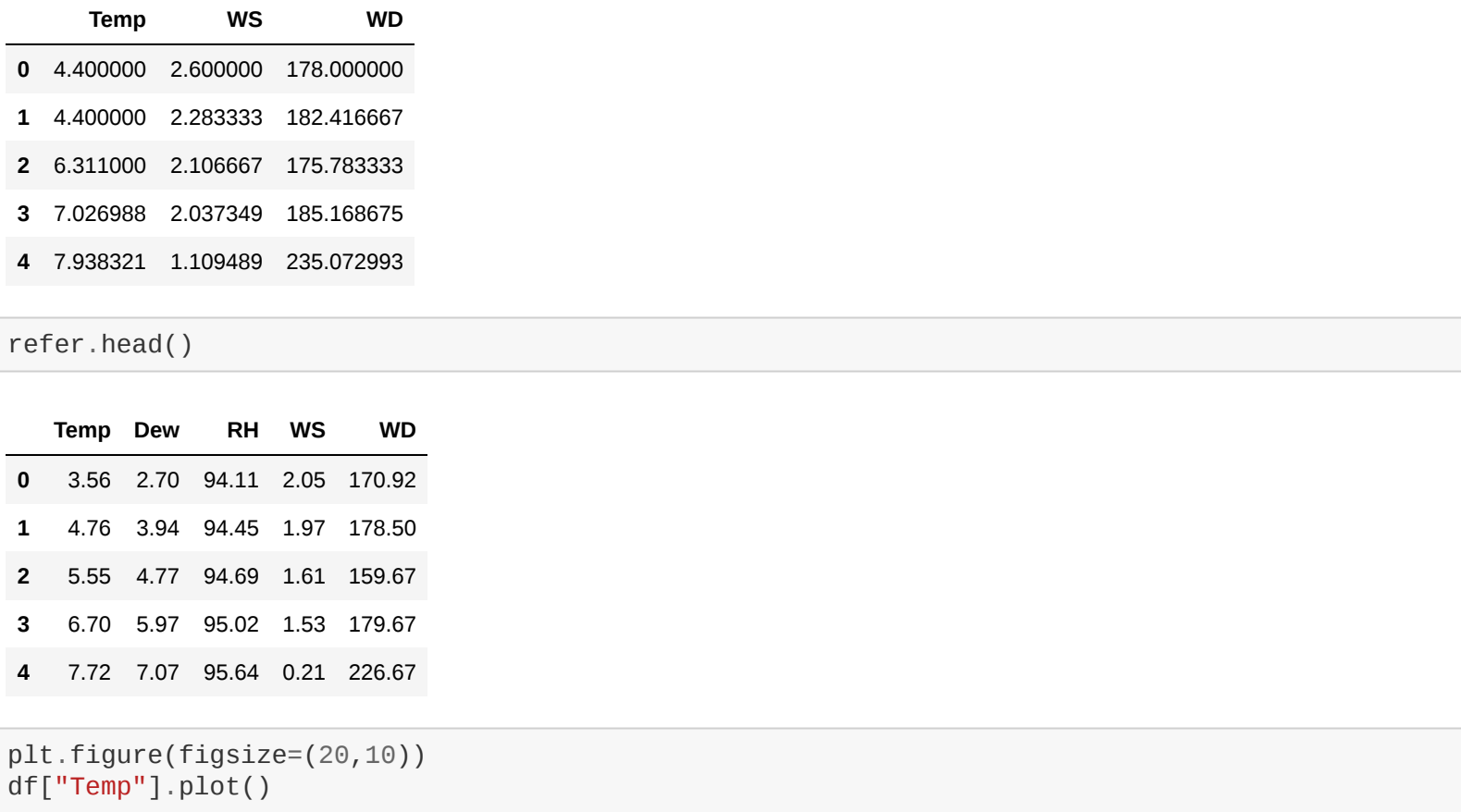
In [25]:
```python
refer.head()
```

Out[25]:

	Temp	Dew	RH	WS	WD
0	3.56	2.70	94.11	2.05	170.92
1	4.76	3.94	94.45	1.97	178.50
2	5.55	4.77	94.69	1.61	159.67
3	6.70	5.97	95.02	1.53	179.67
4	7.72	7.07	95.64	0.21	226.67

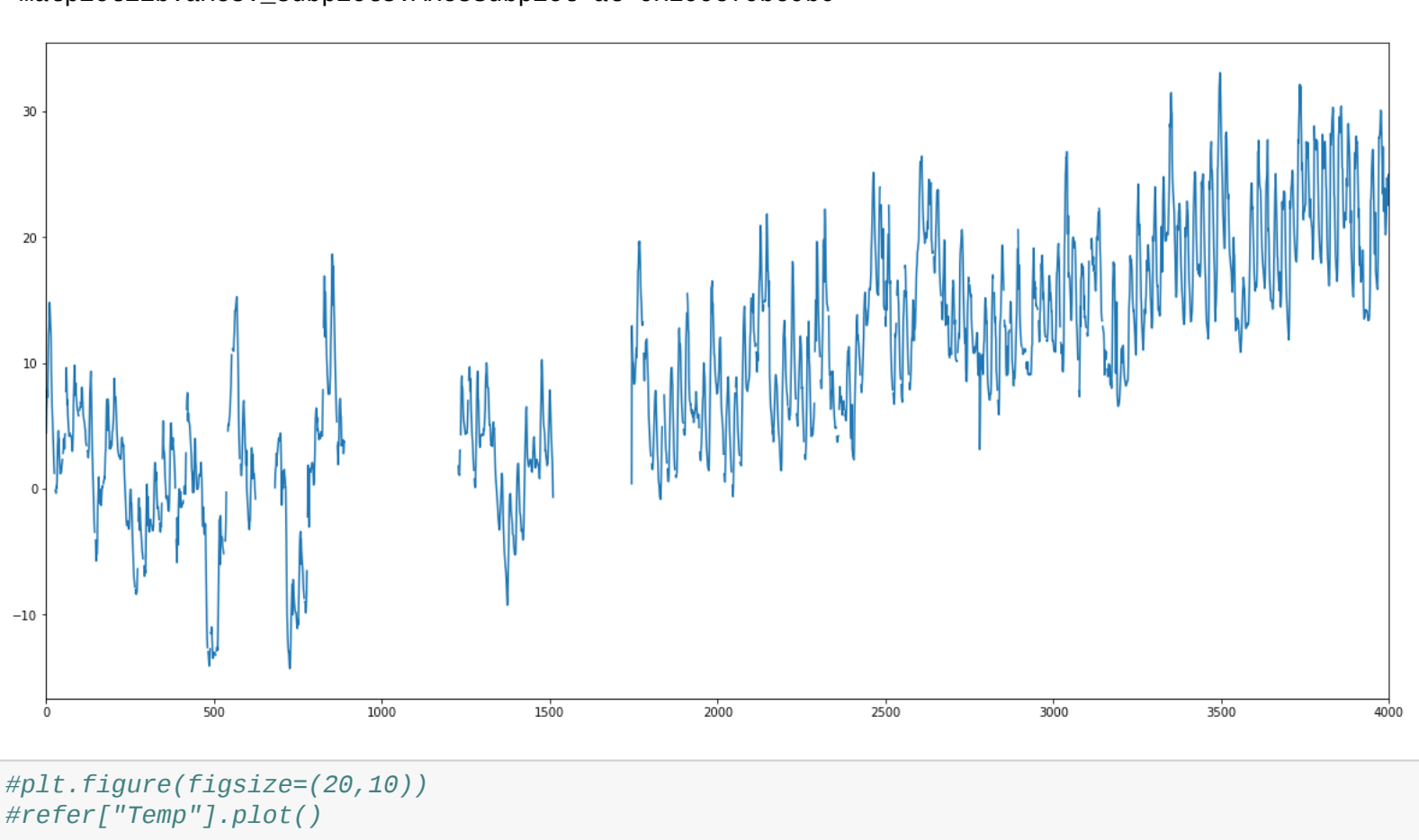
In [26]:
```python
plt.figure(figsize=(20,10))
df["Temp"].plot()
```

Out[26]: <matplotlib.axes._subplots.AxesSubplot at 0x299e79bc9b0>



In [27]:
```python
#plt.figure(figsize=(20,10))
#refer["Temp"].plot()
```

Out[27]: <matplotlib.axes._subplots.AxesSubplot at 0x299e7f4f7b8>



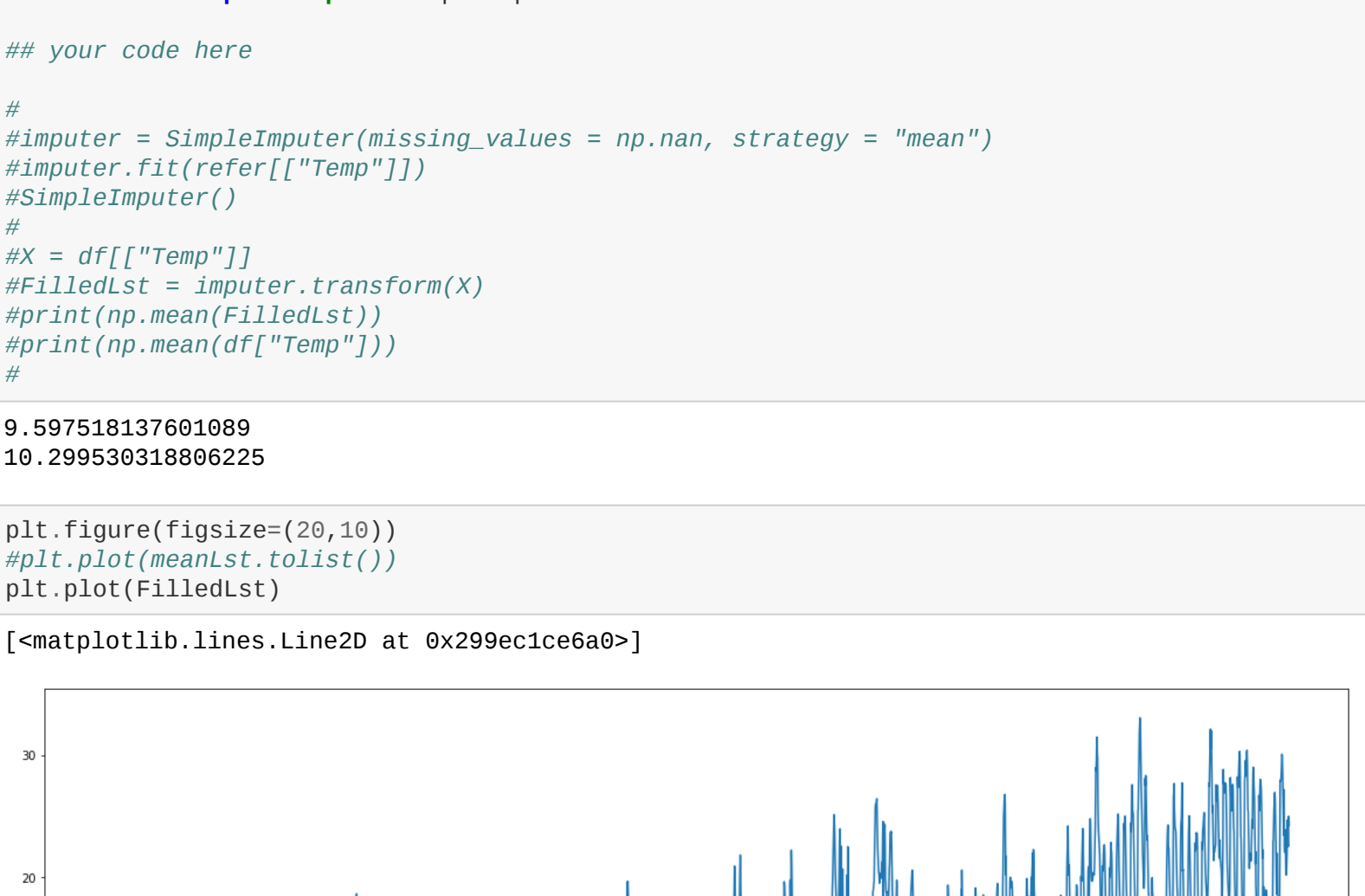
Imputing mean values

In [31]:
```python
from sklearn.impute import SimpleImputer

## your code here

#
#imputer = SimpleImputer(missing_values = np.nan, strategy = "mean")
#imputer.fit(refer[["Temp"]])
#SimpleImputer()
#
#X = df[["Temp"]]
#FilledLst = imputer.transform(X)
#print(np.mean(FilledLst))
#print(np.mean(df["Temp"]))
#
```

```
9.597518137601089
10.299530318806225
```

In [33]:
```python
plt.figure(figsize=(20,10))
#plt.plot(meanLst.tolist())
plt.plot(FilledLst)
```

Out[33]: [<matplotlib.lines.Line2D at 0x299ec1ce6a0>]



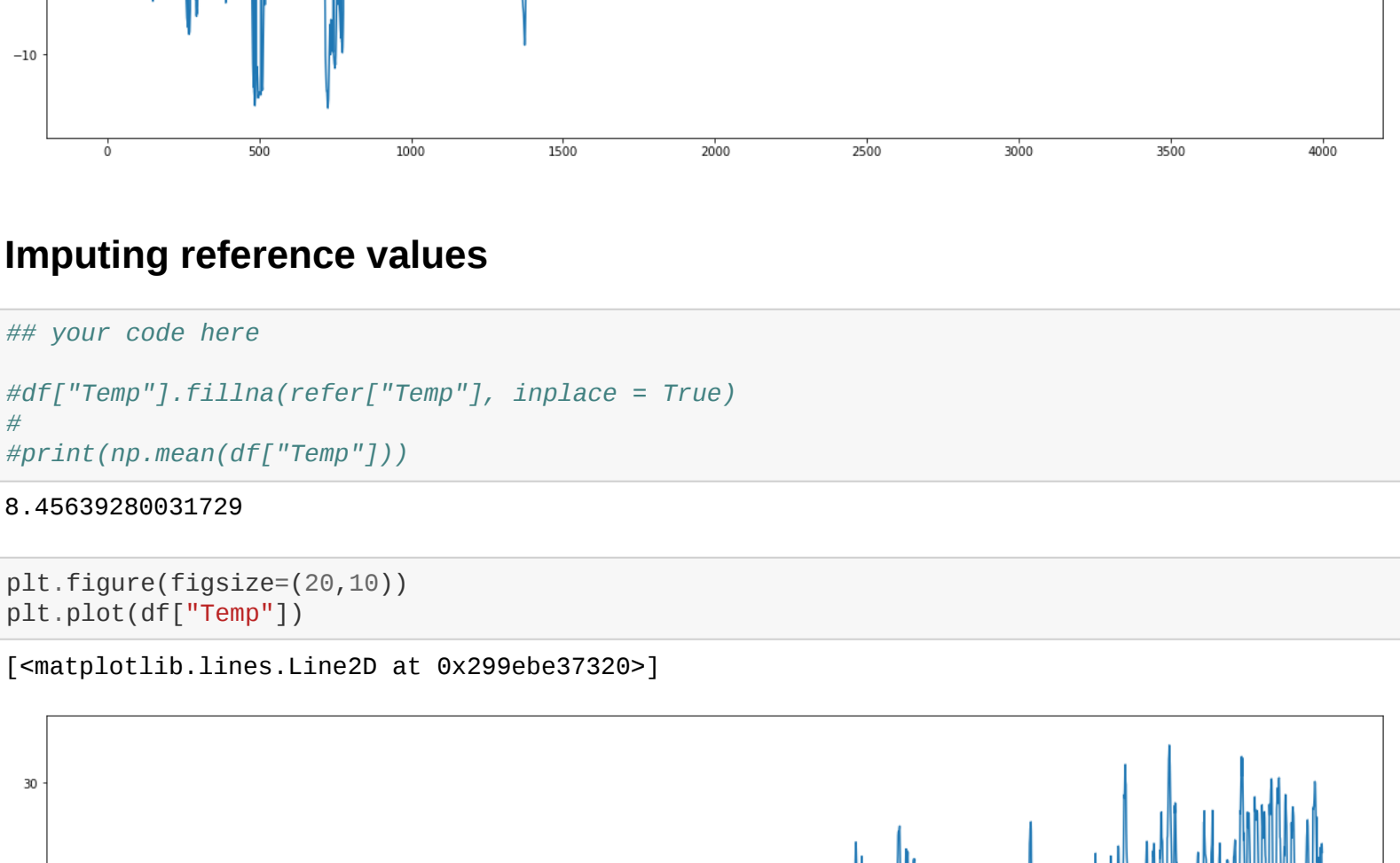
Imputing reference values

In [34]:
```python
## your code here

#df["Temp"].fillna(refer["Temp"], inplace = True)
#
#print(np.mean(df["Temp"]))
```

```
8.45639280031729
```

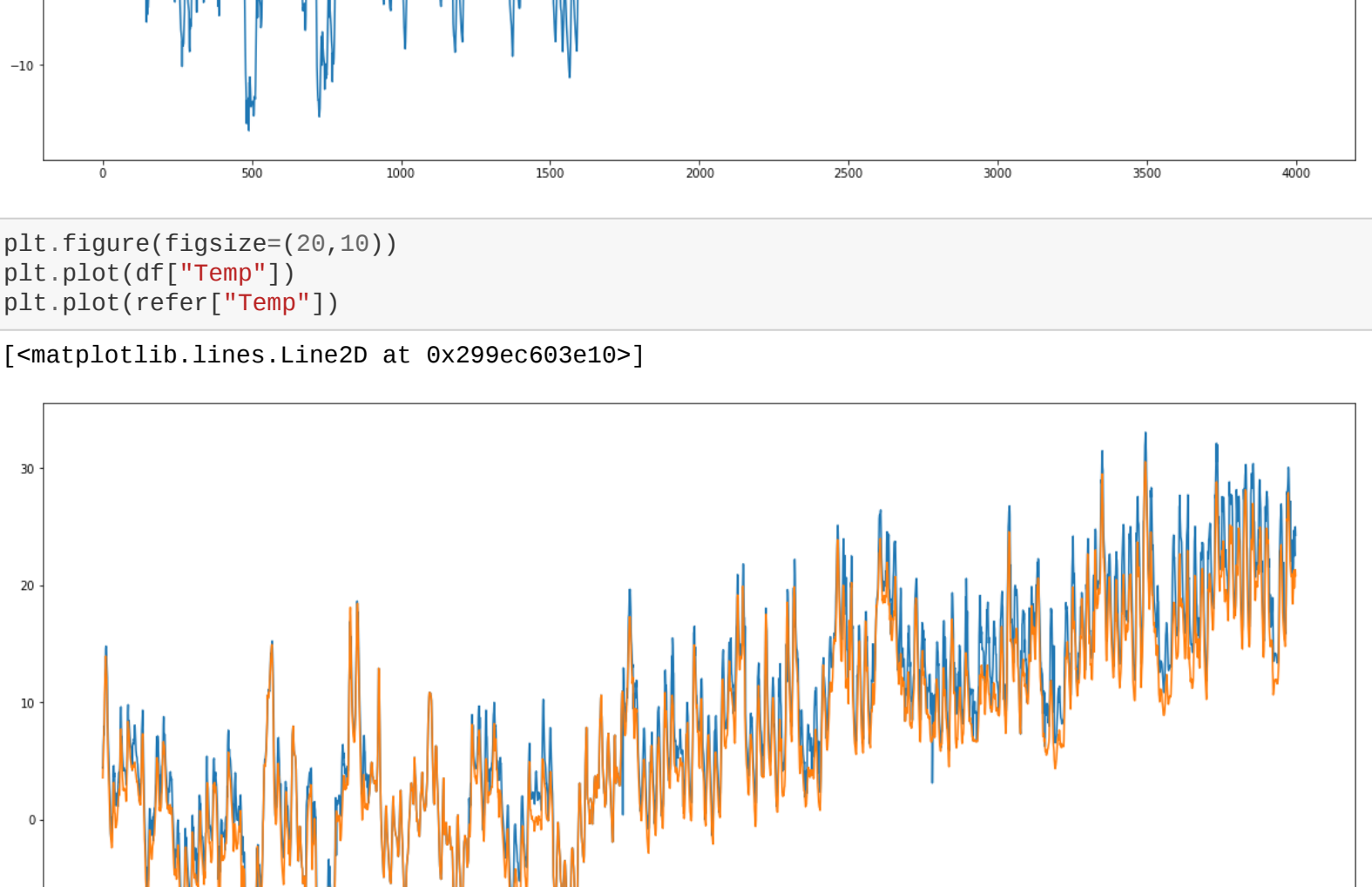
In [38]:
```python
plt.figure(figsize=(20,10))
plt.plot(df["Temp"])
```

Out[38]: [<matplotlib.lines.Line2D at 0x299ebe37320>]



In [35]:
```python
plt.figure(figsize=(20,10))
plt.plot(df["Temp"])
plt.plot(refer["Temp"])
```

Out[35]: [<matplotlib.lines.Line2D at 0x299ec603e10>]



In [ ]: