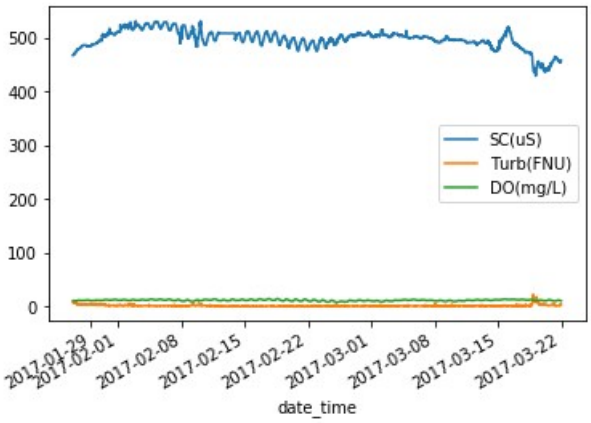
```
2017-01-27 04:00:00   470.0        7.6      10.4
2017-01-27 04:45:00   471.0        7.7      10.4
2017-01-27 05:00:00   471.0        9.8      10.4
2017-01-27 05:30:00   472.0        6.3      10.5
2017-01-27 05:45:00   472.0        7.4      10.5
2017-01-27 06:15:00   472.0        8.0      10.5
2017-01-27 06:30:00   473.0        8.8      10.5
2017-01-27 07:00:00   473.0        6.3      10.5
2017-01-27 07:15:00   474.0        6.1      10.5
2017-01-27 07:30:00   474.0        6.1      10.5
2017-01-27 07:45:00   474.0        5.9      10.5
2017-01-27 08:00:00   474.0        5.8      10.5
2017-01-27 08:30:00   475.0        5.5      10.6
2017-01-27 08:45:00   475.0        6.7      10.6
2017-01-27 09:00:00   475.0        5.4      10.6
2017-01-27 09:15:00   475.0        5.9      10.6
2017-01-27 09:30:00   476.0        5.6      10.7
...                     ...        ...       ...
2017-03-21 16:30:00   457.0        1.8      12.2
2017-03-21 16:45:00   456.0        1.8      12.2
2017-03-21 17:00:00   456.0        2.0      12.2
2017-03-21 17:15:00   456.0        1.8      12.2
2017-03-21 17:30:00   456.0        1.4      12.2
2017-03-21 17:45:00   456.0        1.9      12.1
2017-03-21 18:00:00   456.0        1.8      12.1
2017-03-21 18:15:00   456.0        1.8      12.0
2017-03-21 18:30:00   458.0        1.7      12.0
2017-03-21 18:45:00   456.0        2.0      11.9
2017-03-21 19:00:00   456.0        2.0      11.8
2017-03-21 19:15:00   456.0        2.0      11.8
2017-03-21 19:30:00   457.0        1.9      11.7
2017-03-21 19:45:00   457.0        1.9      11.6
2017-03-21 20:00:00   457.0        2.4      11.5
2017-03-21 20:15:00   457.0        2.2      11.4
2017-03-21 20:30:00   454.0        2.2      11.3
2017-03-21 20:45:00   455.0        2.8      11.3
2017-03-21 21:00:00   455.0        3.0      11.2
2017-03-21 21:15:00   455.0        2.9      11.1
2017-03-21 21:30:00   455.0        3.3      11.1
2017-03-21 21:45:00   454.0        2.9      11.0
2017-03-21 22:00:00   455.0        3.1      10.9
2017-03-21 22:15:00   455.0        2.9      10.9
2017-03-21 22:30:00   455.0        2.8      10.8
2017-03-21 22:45:00   456.0        2.6      10.7
2017-03-21 23:00:00   456.0        2.6      10.7
2017-03-21 23:15:00   456.0        2.6      10.6
2017-03-21 23:30:00   457.0        3.0      10.5
2017-03-21 23:45:00   457.0        3.5      10.5

[4920 rows x 3 columns]

In [12]: dataset
Out[12]:
                     SC(uS)  Turb(FNU)  DO(mg/L)
date_time
```

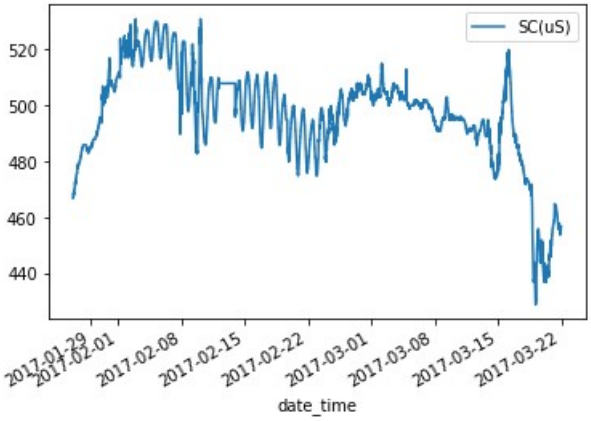
2017-01-27 00:00:00	467.0	8.3	10.4
2017-01-27 00:15:00	467.0	NaN	10.4
2017-01-27 00:30:00	467.0	7.8	10.4
2017-01-27 00:45:00	467.0	8.0	10.4
2017-01-27 01:00:00	468.0	7.5	10.4
2017-01-27 01:15:00	468.0	8.6	10.4
2017-01-27 01:30:00	469.0	8.7	10.4
2017-01-27 01:45:00	468.0	NaN	10.4
2017-01-27 02:00:00	469.0	7.8	10.4
2017-01-27 02:15:00	469.0	8.5	10.4
2017-01-27 02:30:00	469.0	NaN	10.4
2017-01-27 02:45:00	469.0	11.5	10.4
2017-01-27 03:00:00	470.0	11.2	10.4
2017-01-27 03:15:00	470.0	9.1	10.4
2017-01-27 03:30:00	470.0	6.9	10.4
2017-01-27 03:45:00	470.0	7.4	10.4
2017-01-27 04:00:00	470.0	7.6	10.4
2017-01-27 04:15:00	471.0	NaN	10.4
2017-01-27 04:30:00	471.0	NaN	10.4
2017-01-27 04:45:00	471.0	7.7	10.4
2017-01-27 05:00:00	471.0	9.8	10.4
2017-01-27 05:15:00	472.0	NaN	10.5
2017-01-27 05:30:00	472.0	6.3	10.5
2017-01-27 05:45:00	472.0	7.4	10.5
2017-01-27 06:00:00	473.0	NaN	10.5
2017-01-27 06:15:00	472.0	8.0	10.5
2017-01-27 06:30:00	473.0	8.8	10.5
2017-01-27 06:45:00	473.0	NaN	10.5
2017-01-27 07:00:00	473.0	6.3	10.5
2017-01-27 07:15:00	474.0	6.1	10.5
...	...	...	...
2017-03-21 16:30:00	457.0	1.8	12.2
2017-03-21 16:45:00	456.0	1.8	12.2
2017-03-21 17:00:00	456.0	2.0	12.2
2017-03-21 17:15:00	456.0	1.8	12.2
2017-03-21 17:30:00	456.0	1.4	12.2
2017-03-21 17:45:00	456.0	1.9	12.1
2017-03-21 18:00:00	456.0	1.8	12.1
2017-03-21 18:15:00	456.0	1.8	12.0
2017-03-21 18:30:00	458.0	1.7	12.0
2017-03-21 18:45:00	456.0	2.0	11.9
2017-03-21 19:00:00	456.0	2.0	11.8
2017-03-21 19:15:00	456.0	2.0	11.8
2017-03-21 19:30:00	457.0	1.9	11.7
2017-03-21 19:45:00	457.0	1.9	11.6
2017-03-21 20:00:00	457.0	2.4	11.5
2017-03-21 20:15:00	457.0	2.2	11.4
2017-03-21 20:30:00	454.0	2.2	11.3
2017-03-21 20:45:00	455.0	2.8	11.3
2017-03-21 21:00:00	455.0	3.0	11.2
2017-03-21 21:15:00	455.0	2.9	11.1
2017-03-21 21:30:00	455.0	3.3	11.1
2017-03-21 21:45:00	454.0	2.9	11.0
2017-03-21 22:00:00	455.0	3.1	10.9
2017-03-21 22:15:00	455.0	2.9	10.9

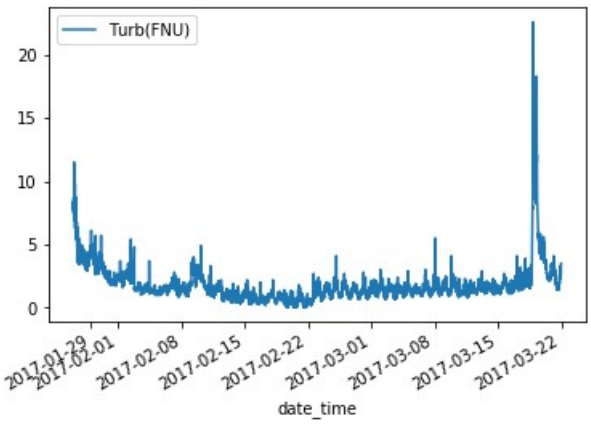
```
2017-03-21 22:30:00   455.0        2.8      10.8
2017-03-21 22:45:00   456.0        2.6      10.7
2017-03-21 23:00:00   456.0        2.6      10.7
2017-03-21 23:15:00   456.0        2.6      10.6
2017-03-21 23:30:00   457.0        3.0      10.5
2017-03-21 23:45:00   457.0        3.5      10.5

[5171 rows x 3 columns]

In [13]: def parser(x):
    ...:     return datetime.strptime(x,'%Y-%m-%d %H:%M')
    ...:
    ...: dataset = pd.read_csv('Data7.csv',header=0, delimiter=',',index_col=0,
parse_dates=[0], date_parser=parser)
    ...: dataset = dataset.fillna(method ='pad')
    ...: #dataset.fillna(method ='bfill')

In [14]: dataset.plot()
    ...: dataset.show()
Traceback (most recent call last):

  File "<ipython-input-14-e3822edd80f4>", line 2, in <module>
    dataset.show()

  File "C:\Users\admin\Anaconda3\lib\site-packages\pandas\core\generic.py", line 4376, in
__getattr__
    return object.__getattribute__(self, name)

AttributeError: 'DataFrame' object has no attribute 'show'
```

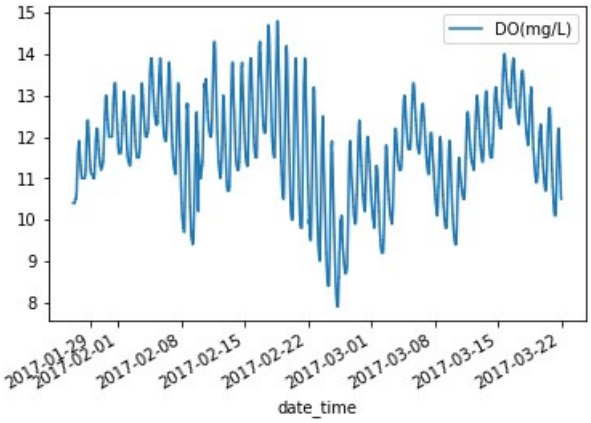



```
In [15]:

In [15]: print('Results of Dickey Fuller Test:')
    ...: dftest = adfuller(dataset['SC(uS)'], autolag='AIC')
    ...:
    ...: dfoutput = pd.Series(dftest[0:4], index=['Test Statistic','p-value','#Lags
Used','Number of Observations Used'])
    ...: for key,value in dftest[4].items():
    ...:     dfoutput['Critical Value (%s)'%key] = value
```

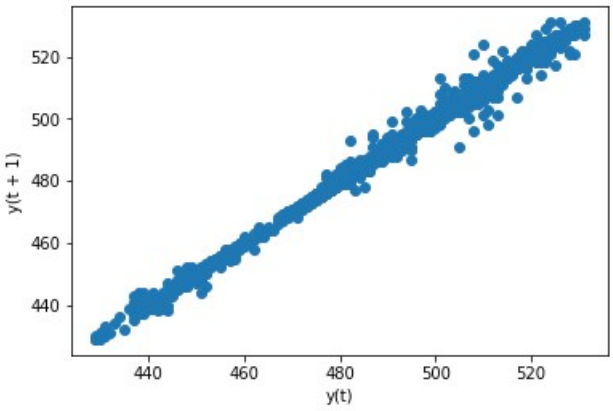
```
    ...:
    ...:
    ...: print(dfoutput)
Results of Dickey Fuller Test:
Test Statistic                  -3.268401
p-value                          0.016351
#Lags Used                      27.000000
Number of Observations Used    5143.000000
Critical Value (1%)              -3.431622
Critical Value (5%)              -2.862102
Critical Value (10%)             -2.567069
dtype: float64

In [16]: sc = dataset.filter(['SC(uS)'], axis=1)
    ...: turb = dataset.filter(['Turb(FNU)'], axis=1)
    ...: do = dataset.filter(['DO(mg/L)'], axis=1)

In [17]: sc.plot()
    ...: turb.plot()
    ...: do.plot()
Out[17]: <matplotlib.axes._subplots.AxesSubplot at 0x228841e6780>
```

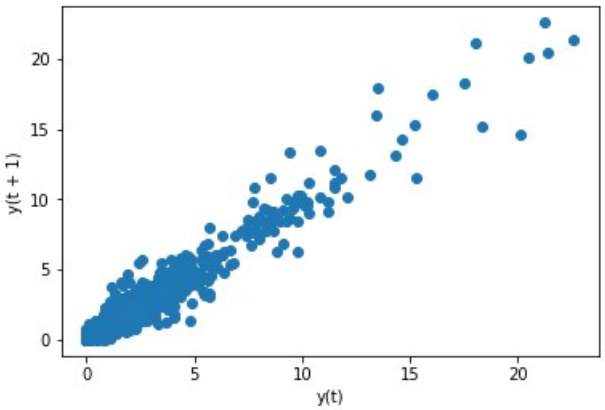


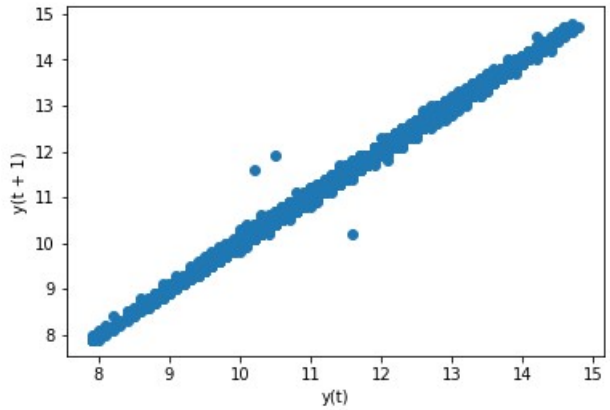





```
In [18]: from pandas.plotting import lag_plot
    ...:
    ...: lag_plot(sc)
    ...: pyplot.show()
    ...:
    ...: lag_plot(turb)
    ...: pyplot.show()
    ...:
    ...: lag_plot(do)
    ...: pyplot.show()
```

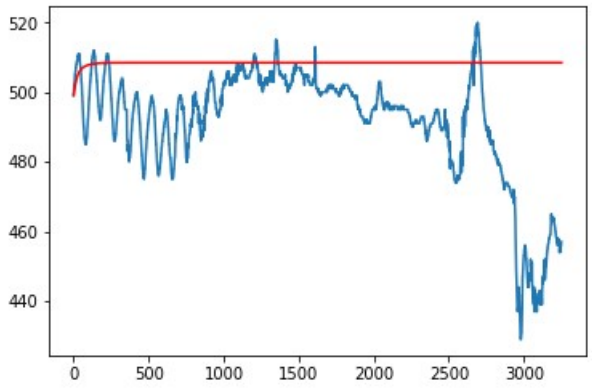







```
In [19]: sc_v = sc.values
    ...: turb_v = turb.values
    ...: do_v = do.values

In [20]: sc_v.size
    ...:
Out[20]: 5171

In [21]: '''
    ...: Training Set- from which the model will learn from
    ...: Test -with which it will compare itself and check itself
    ...:
    ...:
    ...:
    ...: from sklearn.model_selection import train_test_split as tts
    ...: sc_train,sc_test = tts(sc_v,test_size = 0.2, random_state=0)
    ...: do_train,do_test = tts(do_v,test_size = 0.2, random_state=0)
    ...: turb_train,turb_test = tts(turb_v,test_size = 0.2, random_state=0)
    ...: '''
    ...:
    ...: train_size = 1920
    ...: test_size = 3252
    ...: sc_train, sc_test = sc_new[0:train_size,:], sc_new[train_size:len(sc_new),:]
Traceback (most recent call last):

  File "<ipython-input-21-36de0382bd72>", line 15, in <module>
    sc_train, sc_test = sc_new[0:train_size,:], sc_new[train_size:len(sc_new),:]

NameError: name 'sc_new' is not defined


In [22]:

In [22]: '''
    ...: Training Set- from which the model will learn from
    ...: Test -with which it will compare itself and check itself
    ...:
    ...:
    ...:
    ...: from sklearn.model_selection import train_test_split as tts
    ...: sc_train,sc_test = tts(sc_v,test_size = 0.2, random_state=0)
```

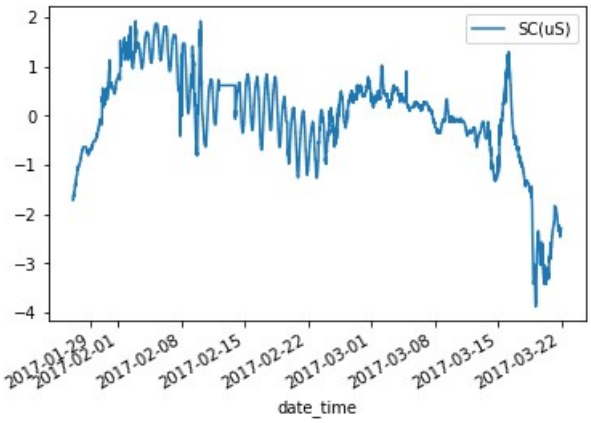
```
    ...: do_train,do_test = tts(do_v,test_size = 0.2, random_state=0)
    ...: turb_train,turb_test = tts(turb_v,test_size = 0.2, random_state=0)
    ...: '''
    ...:
    ...: train_size = 1920
    ...: test_size = 3252
    ...: sc_train, sc_test = sc_v[0:train_size,:], sc_v[train_size:len(sc_v),:]

In [23]: sc_arima = ARIMA(sc_train,order=(12,0,0))
    ...: sc_arima_fit = sc_arima.fit(disp=0)
    ...: print(sc_arima_fit.aic)
C:\Users\admin\Anaconda3\lib\site-packages\scipy\signal\signaltools.py:1341:
FutureWarning: Using a non-tuple sequence for multidimensional indexing is deprecated; use
`arr[tuple(seq)]` instead of `arr[seq]`. In the future this will be interpreted as an
array index, `arr[np.array(seq)]`, which will result either in an error or a different
result.
  out_full[ind] += zi
C:\Users\admin\Anaconda3\lib\site-packages\scipy\signal\signaltools.py:1344:
FutureWarning: Using a non-tuple sequence for multidimensional indexing is deprecated; use
`arr[tuple(seq)]` instead of `arr[seq]`. In the future this will be interpreted as an
array index, `arr[np.array(seq)]`, which will result either in an error or a different
result.
  out = out_full[ind]
C:\Users\admin\Anaconda3\lib\site-packages\scipy\signal\signaltools.py:1350:
FutureWarning: Using a non-tuple sequence for multidimensional indexing is deprecated; use
`arr[tuple(seq)]` instead of `arr[seq]`. In the future this will be interpreted as an
array index, `arr[np.array(seq)]`, which will result either in an error or a different
result.
  zf = out_full[ind]
6524.732499304835

In [24]: predictions= sc_arima_fit.forecast()[0]
    ...: print(predictions)
[499.02522816]

In [25]: from statsmodels.tsa.ar_model import AR
    ...: from sklearn.metrics import mean_squared_error
    ...: sc_ar = AR(sc_train)
    ...: sc_ar_fit = sc_ar.fit()
    ...:
    ...: predictions = sc_ar_fit.predict(start=train_size, end=5171)

In [26]: mp.plot(sc_test)
    ...: mp.plot(predictions,color='red')
Out[26]: [<matplotlib.lines.Line2D at 0x22884bde0b8>]
```



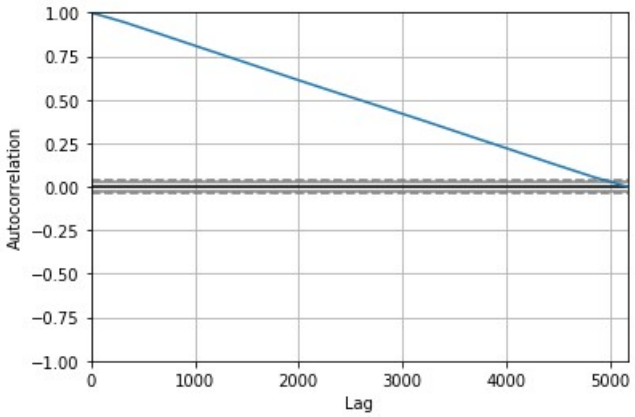

```
In [27]: sc_arima = ARIMA(sc_train,order=(12,0,0))
    ...: sc_arima_fit = sc_arima.fit(disp=0)
    ...: print(sc_arima_fit.aic)
6524.732499304835

In [28]: predictions= sc_arima_fit.forecast(start=train_size, end=5171)[0]
    ...: print(predictions)
Traceback (most recent call last):

  File "<ipython-input-28-4349f0539f7f>", line 1, in <module>
    predictions= sc_arima_fit.forecast(start=train_size, end=5171)[0]

TypeError: forecast() got an unexpected keyword argument 'start'


In [29]:

In [29]: from scipy.stats import zscore
    ...: sc_new = zscore(sc)

In [30]: sc_new.plot()
Traceback (most recent call last):

  File "<ipython-input-30-8a2368c8d0ce>", line 1, in <module>
    sc_new.plot()

AttributeError: 'numpy.ndarray' object has no attribute 'plot'


In [31]:

In [30]:

In [31]: print(sc[0])
Traceback (most recent call last):

  File "<ipython-input-31-0fdf794e35b3>", line 1, in <module>
    print(sc[0])

  File "C:\Users\admin\Anaconda3\lib\site-packages\pandas\core\frame.py", line 2688, in
__getitem__
    return self._getitem_column(key)
```

```
  File "C:\Users\admin\Anaconda3\lib\site-packages\pandas\core\frame.py", line 2695, in
_getitem_column
    return self._get_item_cache(key)

  File "C:\Users\admin\Anaconda3\lib\site-packages\pandas\core\generic.py", line 2489, in
_get_item_cache
    values = self._data.get(item)

  File "C:\Users\admin\Anaconda3\lib\site-packages\pandas\core\internals.py", line 4115,
in get
    loc = self.items.get_loc(item)

  File "C:\Users\admin\Anaconda3\lib\site-packages\pandas\core\indexes\base.py", line
3080, in get_loc
    return self._engine.get_loc(self._maybe_cast_indexer(key))

  File "pandas\_libs\index.pyx", line 140, in pandas._libs.index.IndexEngine.get_loc

  File "pandas\_libs\index.pyx", line 162, in pandas._libs.index.IndexEngine.get_loc

  File "pandas\_libs\hashtable_class_helper.pxi", line 1492, in
pandas._libs.hashtable.PyObjectHashTable.get_item

  File "pandas\_libs\hashtable_class_helper.pxi", line 1500, in
pandas._libs.hashtable.PyObjectHashTable.get_item

KeyError: 0


In [32]:

In [32]: from scipy.stats import zscore
    ...: sc.apply(zscore)
Out[32]:
                     SC(uS)
date_time
2017-01-27 00:00:00 -1.719805
2017-01-27 00:15:00 -1.719805
2017-01-27 00:30:00 -1.719805
2017-01-27 00:45:00 -1.719805
2017-01-27 01:00:00 -1.662872
2017-01-27 01:15:00 -1.662872
2017-01-27 01:30:00 -1.605940
2017-01-27 01:45:00 -1.662872
2017-01-27 02:00:00 -1.605940
2017-01-27 02:15:00 -1.605940
2017-01-27 02:30:00 -1.605940
2017-01-27 02:45:00 -1.605940
2017-01-27 03:00:00 -1.549007
2017-01-27 03:15:00 -1.549007
2017-01-27 03:30:00 -1.549007
2017-01-27 03:45:00 -1.549007
2017-01-27 04:00:00 -1.549007
2017-01-27 04:15:00 -1.492074
```

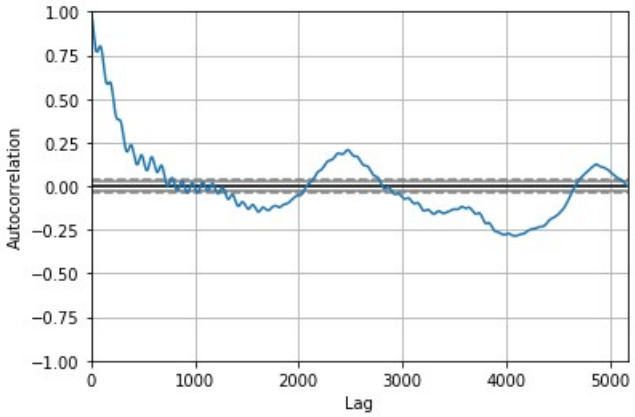
```
2017-01-27 04:30:00 -1.492074
2017-01-27 04:45:00 -1.492074
2017-01-27 05:00:00 -1.492074
2017-01-27 05:15:00 -1.435142
2017-01-27 05:30:00 -1.435142
2017-01-27 05:45:00 -1.435142
2017-01-27 06:00:00 -1.378209
2017-01-27 06:15:00 -1.435142
2017-01-27 06:30:00 -1.378209
2017-01-27 06:45:00 -1.378209
2017-01-27 07:00:00 -1.378209
2017-01-27 07:15:00 -1.321276
...                       ...
2017-03-21 16:30:00 -2.289132
2017-03-21 16:45:00 -2.346064
2017-03-21 17:00:00 -2.346064
2017-03-21 17:15:00 -2.346064
2017-03-21 17:30:00 -2.346064
2017-03-21 17:45:00 -2.346064
2017-03-21 18:00:00 -2.346064
2017-03-21 18:15:00 -2.346064
2017-03-21 18:30:00 -2.232199
2017-03-21 18:45:00 -2.346064
2017-03-21 19:00:00 -2.346064
2017-03-21 19:15:00 -2.346064
2017-03-21 19:30:00 -2.289132
2017-03-21 19:45:00 -2.289132
2017-03-21 20:00:00 -2.289132
2017-03-21 20:15:00 -2.289132
2017-03-21 20:30:00 -2.459930
2017-03-21 20:45:00 -2.402997
2017-03-21 21:00:00 -2.402997
2017-03-21 21:15:00 -2.402997
2017-03-21 21:30:00 -2.402997
2017-03-21 21:45:00 -2.459930
2017-03-21 22:00:00 -2.402997
2017-03-21 22:15:00 -2.402997
2017-03-21 22:30:00 -2.402997
2017-03-21 22:45:00 -2.346064
2017-03-21 23:00:00 -2.346064
2017-03-21 23:15:00 -2.346064
2017-03-21 23:30:00 -2.289132
2017-03-21 23:45:00 -2.289132

[5171 rows x 1 columns]

In [33]: from scipy.stats import zscore
    ...: sc_z = sc.apply(zscore)

In [34]: sc_z.plot()
Out[34]: <matplotlib.axes._subplots.AxesSubplot at 0x22884525208>
```



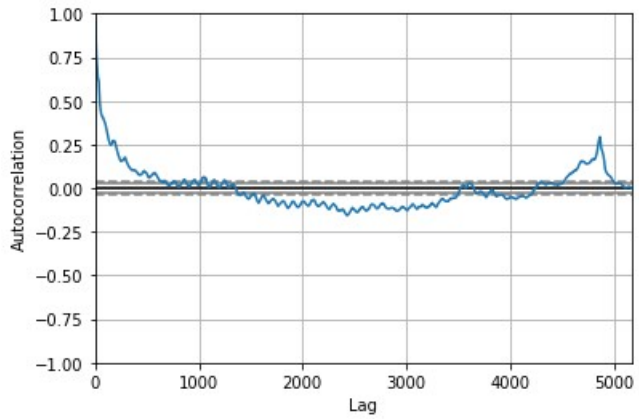

```
In [35]: autocorrelation_plot(dataset)
    ...: pyplot.show()
```



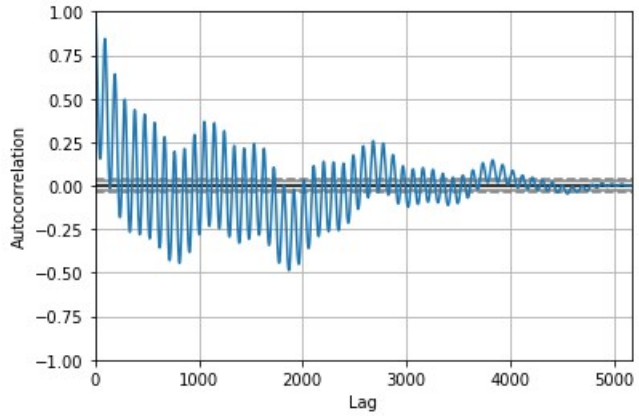

```
In [36]: autocorrelation_plot(sc)
    ...: pyplot.show()
```




```
In [37]: autocorrelation_plot(turb)
    ...: pyplot.show()
```




```
In [38]: autocorrelation_plot(do)
    ...: pyplot.show()
```




```
In [39]: dataset.head()
Out[39]:
                     SC(uS)  Turb(FNU)  DO(mg/L)
date_time
2017-01-27 00:00:00   467.0        8.3      10.4
2017-01-27 00:15:00   467.0        8.3      10.4
2017-01-27 00:30:00   467.0        7.8      10.4
2017-01-27 00:45:00   467.0        8.0      10.4
2017-01-27 01:00:00   468.0        7.5      10.4

In [40]: print('Results of Dickey Fuller Test:')
    ...: dftest = adfuller(dataset['SC(uS)'], autolag='AIC')
    ...:
    ...: dfoutput = pd.Series(dftest[0:4], index=['Test Statistic','p-value','#Lags
Used','Number of Observations Used'])
    ...: for key,value in dftest[4].items():
    ...:     dfoutput['Critical Value (%s)'%key] = value
    ...:
    ...:
    ...: print(dfoutput)
    ...:
    ...: print('Results of Dickey Fuller Test:')
    ...: dftest = adfuller(dataset['Turb(FNU)'], autolag='AIC')
    ...:
```

```
    ...: dfoutput = pd.Series(dftest[0:4], index=['Test Statistic','p-value','#Lags
Used','Number of Observations Used'])
    ...: for key,value in dftest[4].items():
    ...:     dfoutput['Critical Value (%s)'%key] = value
    ...:
    ...:
    ...: print(dfoutput)
    ...:
    ...: print('Results of Dickey Fuller Test:')
    ...: dftest = adfuller(dataset['DO(mg/L)'], autolag='AIC')
    ...:
    ...: dfoutput = pd.Series(dftest[0:4], index=['Test Statistic','p-value','#Lags
Used','Number of Observations Used'])
    ...: for key,value in dftest[4].items():
    ...:     dfoutput['Critical Value (%s)'%key] = value
    ...:
    ...:
    ...: print(dfoutput)
Results of Dickey Fuller Test:
Test Statistic                   -3.268401
p-value                           0.016351
#Lags Used                       27.000000
Number of Observations Used    5143.000000
Critical Value (1%)              -3.431622
Critical Value (5%)              -2.862102
Critical Value (10%)             -2.567069
dtype: float64
Results of Dickey Fuller Test:
Test Statistic                   -5.039398
p-value                           0.000019
#Lags Used                       31.000000
Number of Observations Used    5139.000000
Critical Value (1%)              -3.431623
Critical Value (5%)              -2.862103
Critical Value (10%)             -2.567069
dtype: float64
Results of Dickey Fuller Test:
Test Statistic                -5.782591e+00
p-value                        5.085067e-07
#Lags Used                     3.200000e+01
Number of Observations Used    5.138000e+03
Critical Value (1%)           -3.431623e+00
Critical Value (5%)           -2.862103e+00
Critical Value (10%)          -2.567070e+00
dtype: float64

In [41]:
```