In [2]:
```
import pandas as pd
df = pd.read_csv("https://raw.githubusercontent.com/KlrShaK/Delhi-Temperature-Prediction-Time-Series-data/master/daily_temp_dataset/DailyDelhiClimateTrain.csv")
df
```

Out[2]:

	date	meantemp	humidity	wind_speed	meanpressure
0	2013-01-01	10.000000	84.500000	0.000000	1015.666667
1	2013-01-02	7.400000	92.000000	2.980000	1017.800000
2	2013-01-03	7.166667	87.000000	4.633333	1018.666667
3	2013-01-04	8.666667	71.333333	1.233333	1017.166667
4	2013-01-05	6.000000	86.833333	3.700000	1016.500000
...	...	...	...	...	...
1457	2016-12-28	17.217391	68.043478	3.547826	1015.565217
1458	2016-12-29	15.238095	87.857143	6.000000	1016.904762
1459	2016-12-30	14.095238	89.666667	6.266667	1017.904762
1460	2016-12-31	15.052632	87.000000	7.325000	1016.100000
1461	2017-01-01	10.000000	100.000000	0.000000	1016.000000

1462 rows × 5 columns

In [11]:
```
df_useful = df.drop(["humidity","wind_speed","meanpressure","date"],axis=1)
df_useful
```

Out[11]:

	meantemp
0	10.000000
1	7.400000
2	7.166667
3	8.666667
4	6.000000
...	...
1457	17.217391
1458	15.238095
1459	14.095238
1460	15.052632
1461	10.000000

1462 rows × 1 columns

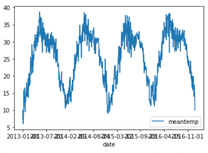
In [12]:
```
df_useful_indexed = df_useful.set_index(df.date)

# df_useful_indexed = pd.to_datetime(df_useful_indexed.index)
df_useful_indexed
```

Out[12]:

	meantemp
date	
2013-01-01	10.000000
2013-01-02	7.400000
2013-01-03	7.166667
2013-01-04	8.666667
2013-01-05	6.000000
...	...
2016-12-28	17.217391
2016-12-29	15.238095
2016-12-30	14.095238
2016-12-31	15.052632
2017-01-01	10.000000

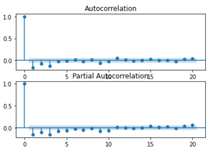
1462 rows × 1 columns

In [15]:
```
from statsmodels.tsa.stattools import adfuller
import matplotlib.pyplot as plt
# Compute and print ADF p-value
df_useful_indexed.plot()
plt.show()
result = adfuller(df_useful_indexed)
print("The p-value for the ADF test is ", result[1])
```




```
The p-value for the ADF test is  0.2774121372301609
```

In [19]:
```
# Import the modules for plotting the sample ACF and PACF
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf

# Take first difference of the temperature Series
chg_temp = df_useful_indexed.diff()
chg_temp = chg_temp.dropna()
# Plot the ACF and PACF on the same page
fig, axes = plt.subplots(2,1)

# Plot the ACF
plot_acf(chg_temp, lags=20, ax=axes[0])

# Plot the PACF
plot_pacf(chg_temp, lags=20, ax=axes[1])
plt.show()
```

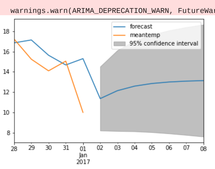


In [20]:
```
# Import the module for estimating an ARMA model
from statsmodels.tsa.arima_model import ARMA

# Fit the data to an AR(1) model and print AIC:
mod_ar1 = ARMA(chg_temp, order=(1, 0))
res_ar1 = mod_ar1.fit()
print("The AIC for an AR(1) is: ", res_ar1.aic)

# Fit the data to an AR(2) model and print AIC:
mod_ar2 = ARMA(chg_temp, order=(2, 0))
res_ar2 = mod_ar2.fit()
print("The AIC for an AR(2) is: ", res_ar2.aic)

# Fit the data to an ARMA(1,1) model and print AIC:
mod_arma11 = ARMA(chg_temp, order=(1, 1))
res_arma11 = mod_arma11.fit()
print("The AIC for an ARMA(1,1) is: ", res_arma11.aic)
```

```
C:\Users\U.R Computer\Anaconda\lib\site-packages\statsmodels\tsa\arima_model.py:472: FutureWarning:
statsmodels.tsa.arima_model.ARMA and statsmodels.tsa.arima_model.ARIMA have
been deprecated in favor of statsmodels.tsa.arima.model.ARIMA (note the .
between arima and model) and
statsmodels.tsa.SARIMAX. These will be removed after the 0.12 release.

statsmodels.tsa.arima.model.ARIMA makes use of the statespace framework and
is both well tested and maintained.

To silence this warning and continue using ARMA and ARIMA until they are
removed, use:

import warnings
warnings.filterwarnings('ignore', 'statsmodels.tsa.arima_model.ARMA',
                        FutureWarning)
warnings.filterwarnings('ignore', 'statsmodels.tsa.arima_model.ARIMA',
                        FutureWarning)

  warnings.warn(ARIMA_DEPRECATION_WARN, FutureWarning)
C:\Users\U.R Computer\Anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:524: ValueWarning: No frequency information was provided, so inferred frequ
ency D will be used.
  warnings.warn('No frequency information was'
The AIC for an AR(1) is:  5614.068005431916
C:\Users\U.R Computer\Anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:524: ValueWarning: No frequency information was provided, so inferred frequ
ency D will be used.
  warnings.warn('No frequency information was'
The AIC for an AR(2) is:  5600.849349106674
C:\Users\U.R Computer\Anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:524: ValueWarning: No frequency information was provided, so inferred frequ
ency D will be used.
  warnings.warn('No frequency information was'
The AIC for an ARMA(1,1) is:  5544.459909968323
```

In [28]:
```
# Import the ARIMA module from statsmodels
from statsmodels.tsa.arima_model import ARIMA

# Forecast temperatures using an ARIMA(1,1,1) model
mod = ARIMA(df_useful_indexed, order=(1,1,1))
forc = mod.fit()

# Plot the original series and the forecasted series
forc.plot_predict(start='2016-12-28', end='2017-1-8')
plt.show()
```

```
C:\Users\U.R Computer\Anaconda\lib\site-packages\statsmodels\tsa\arima_model.py:472: FutureWarning:
statsmodels.tsa.arima_model.ARMA and statsmodels.tsa.arima_model.ARIMA have
been deprecated in favor of statsmodels.tsa.arima.model.ARIMA (note the .
between arima and model) and
statsmodels.tsa.SARIMAX. These will be removed after the 0.12 release.

statsmodels.tsa.arima.model.ARIMA makes use of the statespace framework and
is both well tested and maintained.

To silence this warning and continue using ARMA and ARIMA until they are
removed, use:

import warnings
warnings.filterwarnings('ignore', 'statsmodels.tsa.arima_model.ARMA',
                        FutureWarning)
warnings.filterwarnings('ignore', 'statsmodels.tsa.arima_model.ARIMA',
                        FutureWarning)

  warnings.warn(ARIMA_DEPRECATION_WARN, FutureWarning)
C:\Users\U.R Computer\Anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:524: ValueWarning: No frequency information was provided, so inferred frequ
ency D will be used.
  warnings.warn('No frequency information was'
C:\Users\U.R Computer\Anaconda\lib\site-packages\statsmodels\tsa\base\tsa_model.py:524: ValueWarning: No frequency information was provided, so inferred frequ
ency D will be used.
  warnings.warn('No frequency information was'
C:\Users\U.R Computer\Anaconda\lib\site-packages\statsmodels\tsa\arima_model.py:472: FutureWarning:
statsmodels.tsa.arima_model.ARMA and statsmodels.tsa.arima_model.ARIMA have
been deprecated in favor of statsmodels.tsa.arima.model.ARIMA (note the .
between arima and model) and
statsmodels.tsa.SARIMAX. These will be removed after the 0.12 release.

statsmodels.tsa.arima.model.ARIMA makes use of the statespace framework and
is both well tested and maintained.

To silence this warning and continue using ARMA and ARIMA until they are
removed, use:

import warnings
warnings.filterwarnings('ignore', 'statsmodels.tsa.arima_model.ARMA',
                        FutureWarning)
warnings.filterwarnings('ignore', 'statsmodels.tsa.arima_model.ARIMA',
                        FutureWarning)

  warnings.warn(ARIMA_DEPRECATION_WARN, FutureWarning)
```



In [ ]: