In [30]:
```
# ! pip install numpy
```

In [37]:
```
import numpy as np
import pandas as pd
import statsmodels as sm
import matplotlib.pyplot as plt
%matplotlib inline
```

In [36]:
```
# from statsmodels.tsa.seasonal import seasonal_decompose
```

In [38]:
```
sm.tsa.seasonal_decompose()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Input In [38], in <cell line: 1>()
----> 1 sm.tsa.seasonal_decompose()

AttributeError: module 'statsmodels' has no attribute 'tsa'
```

In [6]:
```
df = pd.read_csv('monthly-milk-production-pounds-p.csv')
```

In [7]:
```
df.head()
```

Out[7]:

| | Month | Monthly milk production: pounds per cow. Jan 62 ? Dec 75 |
|---|---|---|
| **0** | 1962-01 | 589.0 |
| **1** | 1962-02 | 561.0 |
| **2** | 1962-03 | 640.0 |
| **3** | 1962-04 | 656.0 |
| **4** | 1962-05 | 727.0 |

In [8]:
```
df.tail()
```

Out[8]:

| | Month | Monthly milk production: pounds per cow. Jan 62 ? Dec 75 |
|---|---|---|
| **164** | 1975-09 | 817.0 |
| **165** | 1975-10 | 827.0 |
| **166** | 1975-11 | 797.0 |
| **167** | 1975-12 | 843.0 |
| **168** | Monthly milk production: pounds per cow. Jan 6... | NaN |

In [9]:
```
df.columns = ['Month', 'Milk in pounds per cow']
```

In [10]:
```
df.head()
```

Out[10]:

| | Month | Milk in pounds per cow |
|---|---|---|
| **0** | 1962-01 | 589.0 |
| **1** | 1962-02 | 561.0 |
| **2** | 1962-03 | 640.0 |
| **3** | 1962-04 | 656.0 |
| **4** | 1962-05 | 727.0 |

In [11]:
```
df.drop(168,axis = 0,inplace = True)
```

In [12]:
```
df['Month'] = pd.to_datetime(df['Month'])
```

In [13]:
```
df['Month']
```

Out[13]:
```
0     1962-01-01
1     1962-02-01
2     1962-03-01
3     1962-04-01
4     1962-05-01
         ...
163   1975-08-01
164   1975-09-01
165   1975-10-01
166   1975-11-01
167   1975-12-01
Name: Month, Length: 168, dtype: datetime64[ns]
```

In [14]:
```
df.set_index('Month',inplace=True)
```

In [15]:
```
df
```

Out[15]:

| Month | Milk in pounds per cow |
|---|---|
| **1962-01-01** | 589.0 |
| **1962-02-01** | 561.0 |
| **1962-03-01** | 640.0 |
| **1962-04-01** | 656.0 |
| **1962-05-01** | 727.0 |
| **...** | ... |
| **1975-08-01** | 858.0 |
| **1975-09-01** | 817.0 |
| **1975-10-01** | 827.0 |
| **1975-11-01** | 797.0 |
| **1975-12-01** | 843.0 |

168 rows × 1 columns

In [16]:
```
df.head()
```

Out[16]:

| Month | Milk in pounds per cow |
|---|---|
| **1962-01-01** | 589.0 |
| **1962-02-01** | 561.0 |
| **1962-03-01** | 640.0 |
| **1962-04-01** | 656.0 |
| **1962-05-01** | 727.0 |

In [17]:
```
df.describe().transpose()
```

Out[17]:

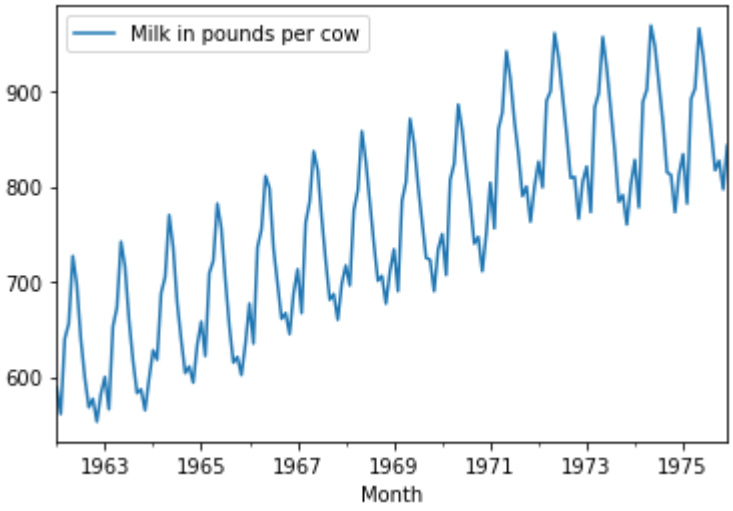
| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| **Milk in pounds per cow** | 168.0 | 754.708333 | 102.204524 | 553.0 | 677.75 | 761.0 | 824.5 | 969.0 |

In [18]:
```
df.index
```

Out[18]:
```
DatetimeIndex(['1962-01-01', '1962-02-01', '1962-03-01', '1962-04-01',
               '1962-05-01', '1962-06-01', '1962-07-01', '1962-08-01',
               '1962-09-01', '1962-10-01',
               ...
               '1975-03-01', '1975-04-01', '1975-05-01', '1975-06-01',
               '1975-07-01', '1975-08-01', '1975-09-01', '1975-10-01',
               '1975-11-01', '1975-12-01'],
              dtype='datetime64[ns]', name='Month', length=168, freq=None)
```

In [19]:
```
df.plot()
```

Out[19]:
```
<AxesSubplot:xlabel='Month'>
```



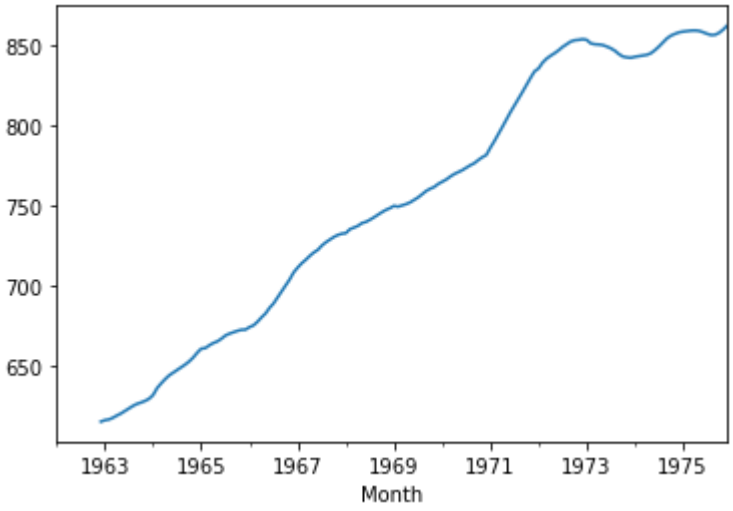

In [20]:
```
timeseries = df['Milk in pounds per cow']
```

In [21]:
```
type(timeseries)
```

Out[21]:
```
pandas.core.series.Series
```

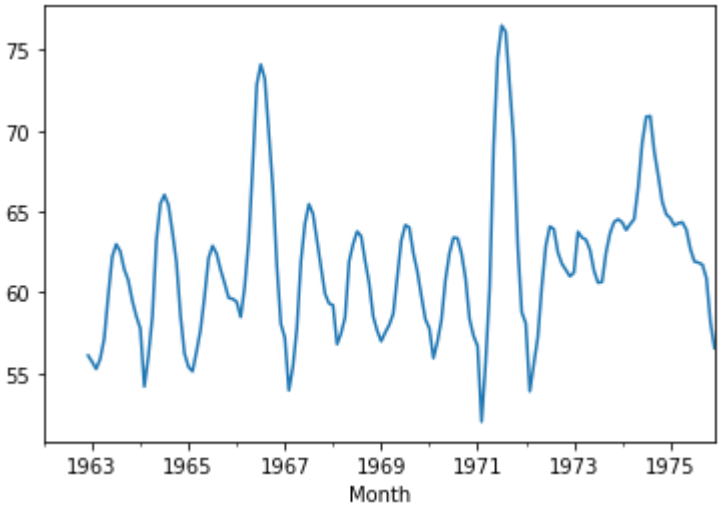
In [22]:
```
timeseries.rolling(12).mean().plot(label='12 Month Rolling Mean')
```

Out[22]:
```
<AxesSubplot:xlabel='Month'>
```



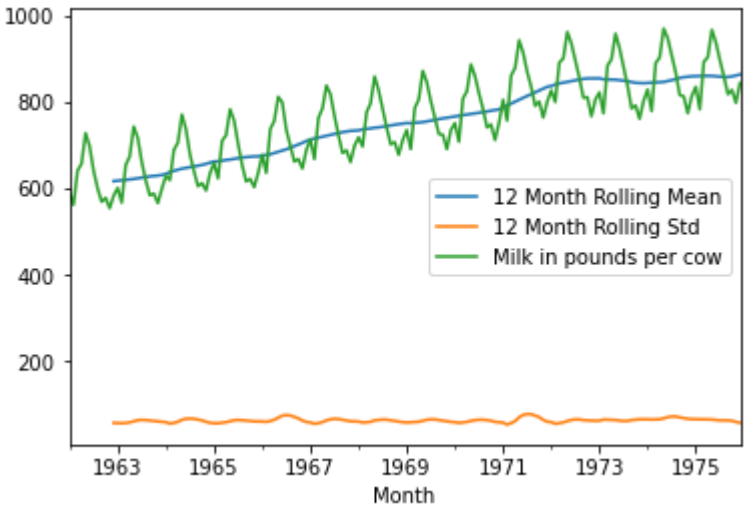
In [23]:
```
timeseries.rolling(12).std().plot(label='12 Month Rolling Std')
```

Out[23]: <AxesSubplot:xlabel='Month'>



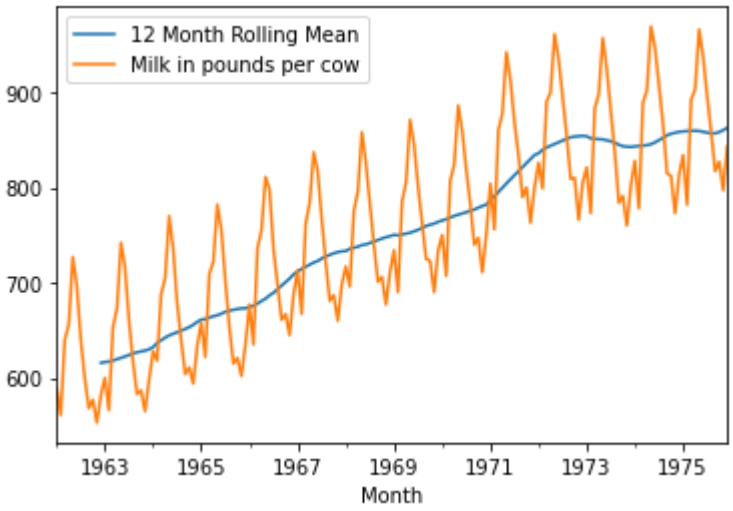

In [24]:
```
timeseries.rolling(12).mean().plot(label='12 Month Rolling Mean')
timeseries.rolling(12).std().plot(label='12 Month Rolling Std')
timeseries.plot()
plt.legend()
```

Out[24]: <matplotlib.legend.Legend at 0x1db3673d040>




In [25]:
```
timeseries.rolling(12).mean().plot(label = '12 Month Rolling Mean')
timeseries.plot()
plt.legend()
```

Out[25]: <matplotlib.legend.Legend at 0x1db3664fbb0>




In [ ]:
```
from statsmodels.tsa.seasonal import seasonal_decompose
```

In [ ]:
```
decomposition = seasonal_decompose(df['Milk in pounds per cow'],freq = 12)
```

In [27]:
```
fig = plt.figure()
fig = decomposition.plot()
fig.set_size_inches(15, 8)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Input In [27], in <cell line: 2>()
      1 fig = plt.figure()
----> 2 fig = decomposition.plot()
      3 fig.set_size_inches(15, 8)

NameError: name 'decomposition' is not defined
<Figure size 432x288 with 0 Axes>
```

In [28]:
```
df.head()
```

Out[28]:

| | Milk in pounds per cow |
|---|---|
| **Month** | |
| **1962-01-01** | 589.0 |
| **1962-02-01** | 561.0 |
| **1962-03-01** | 640.0 |
| **1962-04-01** | 656.0 |
| **1962-05-01** | 727.0 |

In [42]:
```
from statsmodels.tsa.stattools import adfuller
```

In [43]:
```
result = adfuller(df['Milk in pounds per cow'])
```

In [44]:
```
labels = ['ADF Test Statistic',
          'p-value',
          '#Lags Used',
          'Number of Observations Used']
```

In [45]:
```
labels
```

Out[45]:
```
['ADF Test Statistic', 'p-value', '#Lags Used', 'Number of Observations Use
d']
```

In [46]:
```
for value,label in zip(result,labels):
    print(label+' : '+str(value) )
```

```
ADF Test Statistic : -1.3038115874221248
p-value : 0.6274267086030337
#Lags Used : 13
Number of Observations Used : 154
```

In [47]:
```
if result[1] <= 0.05:
    print("strong evidence against the null hypothesis, reject the null hypo
else:
    print("weak evidence against null hypothesis, time series has a unit roo
```

```
weak evidence against null hypothesis, time series has a unit root, indicatin
g it is non-stationary
```

In [48]:
```
def adf_check(time_series):
    """
    Pass in a time series, returns ADF report
    """
    result = adfuller(time_series)
    print('Augmented Dickey-Fuller Test:')
    labels = ['ADF Test Statistic',
              'p-value',
              '#Lags Used',
              'Number of Observations Used']

    for value,label in zip(result,labels):
        print(label+' : '+str(value) )

    if result[1] <= 0.05:
        print("strong evidence against the null hypothesis, reject the null
    else:
        print("weak evidence against null hypothesis, time series has a unit
```

In [49]:
```
df['Milk First Difference'] = df['Milk in pounds per cow'] - df['Milk in pou
```

In [50]:
```
df['Milk First Difference']
```

Out[50]:
```
Month
1962-01-01     NaN
1962-02-01   -28.0
1962-03-01    79.0
1962-04-01    16.0
1962-05-01    71.0
1962-06-01   -30.0
1962-07-01   -57.0
1962-08-01   -41.0
1962-09-01   -31.0
1962-10-01     9.0
1962-11-01   -24.0
1962-12-01    29.0
1963-01-01    18.0
1963-02-01   -34.0
1963-03-01    87.0
1963-04-01    20.0
1963-05-01    69.0
1963-06-01   -26.0
1963-07-01   -56.0
1963-08-01   -43.0
1963-09-01   -34.0
1963-10-01     4.0
1963-11-01   -22.0
1963-12-01    33.0
1964-01-01    30.0
1964-02-01   -10.0
1964-03-01    70.0
1964-04-01    17.0
1964-05-01    65.0
1964-06-01   -34.0
              ...
1973-07-01   -43.0
1973-08-01   -44.0
1973-09-01   -53.0
1973-10-01     7.0
1973-11-01   -31.0
1973-12-01    42.0
1974-01-01    26.0
1974-02-01   -50.0
1974-03-01   111.0
1974-04-01    13.0
1974-05-01    67.0
1974-06-01   -22.0
1974-07-01   -39.0
1974-08-01   -41.0
1974-09-01   -52.0
1974-10-01    -3.0
1974-11-01   -39.0
1974-12-01    40.0
1975-01-01    21.0
1975-02-01   -52.0
1975-03-01   110.0
1975-04-01    11.0
1975-05-01    63.0
1975-06-01   -29.0
```

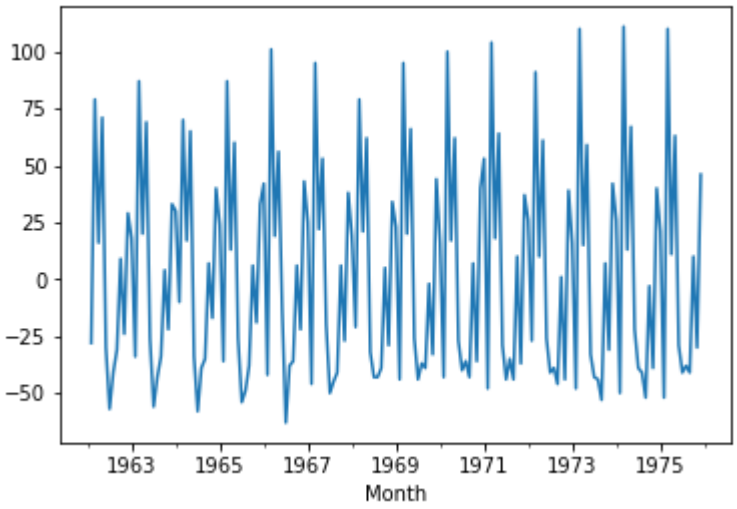
```
1975-07-01    -41.0
1975-08-01    -38.0
1975-09-01    -41.0
1975-10-01     10.0
1975-11-01    -30.0
1975-12-01     46.0
Name: Milk First Difference, Length: 168, dtype: float64
```

In [51]:
```
adf_check(df['Milk First Difference'].dropna())
```

```
Augmented Dickey-Fuller Test:
ADF Test Statistic : -3.05499555865311
p-value : 0.030068004001782334
#Lags Used : 14
Number of Observations Used : 152
strong evidence against the null hypothesis, reject the null hypothesis. Data
has no unit root and is stationary
```

In [52]:
```
df['Milk First Difference'].plot()
```

Out[52]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2988060be0>
```




In [53]:
```
df['Seasonal Difference'] = df['Milk in pounds per cow'] - df['Milk in pounc
```

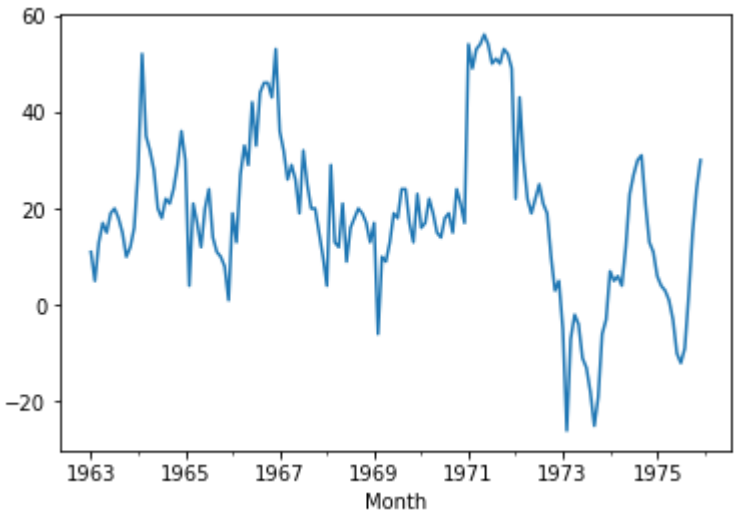
In [54]:
```
df['Seasonal Difference']
```

Out[54]:
```
Month
1962-01-01     NaN
1962-02-01     NaN
1962-03-01     NaN
1962-04-01     NaN
1962-05-01     NaN
1962-06-01     NaN
1962-07-01     NaN
1962-08-01     NaN
1962-09-01     NaN
1962-10-01     NaN
1962-11-01     NaN
1962-12-01     NaN
1963-01-01    11.0
1963-02-01     5.0
1963-03-01    13.0
1963-04-01    17.0
1963-05-01    15.0
1963-06-01    19.0
1963-07-01    20.0
1963-08-01    18.0
1963-09-01    15.0
1963-10-01    10.0
1963-11-01    12.0
1963-12-01    16.0
1964-01-01    28.0
1964-02-01    52.0
1964-03-01    35.0
1964-04-01    32.0
1964-05-01    28.0
1964-06-01    20.0
              ...
1973-07-01   -13.0
1973-08-01   -18.0
1973-09-01   -25.0
1973-10-01   -19.0
1973-11-01    -6.0
1973-12-01    -3.0
1974-01-01     7.0
1974-02-01     5.0
1974-03-01     6.0
1974-04-01     4.0
1974-05-01    12.0
1974-06-01    23.0
1974-07-01    27.0
1974-08-01    30.0
1974-09-01    31.0
1974-10-01    21.0
1974-11-01    13.0
1974-12-01    11.0
1975-01-01     6.0
1975-02-01     4.0
1975-03-01     3.0
1975-04-01     1.0
1975-05-01    -3.0
1975-06-01   -10.0
```

```
1975-07-01   -12.0
1975-08-01    -9.0
1975-09-01     2.0
1975-10-01    15.0
1975-11-01    24.0
1975-12-01    30.0
Name: Seasonal Difference, Length: 168, dtype: float64
```

In [55]:
```
df['Seasonal Difference'].plot()
```

Out[55]: <matplotlib.axes._subplots.AxesSubplot at 0x7f299ec80a20>



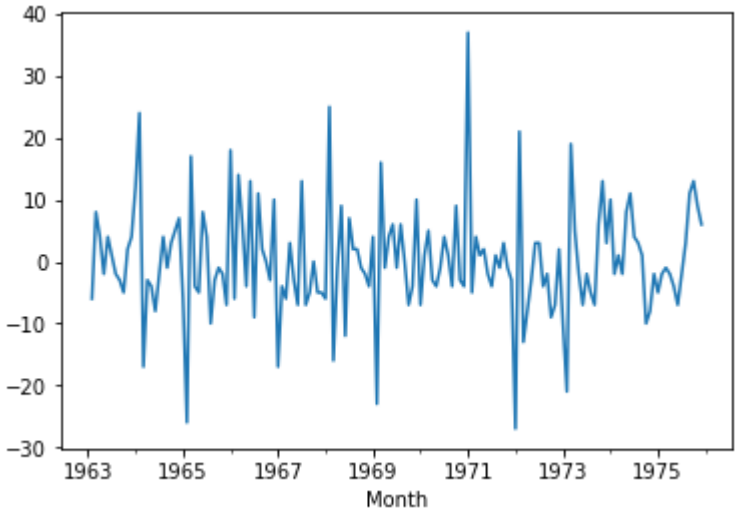

In [56]:
```
adf_check(df['Seasonal Difference'].dropna())
```

```
Augmented Dickey-Fuller Test:
ADF Test Statistic : -2.335419314359397
p-value : 0.16079880527711382
#Lags Used : 12
Number of Observations Used : 143
weak evidence against null hypothesis, time series has a unit root, indicatin
g it is non-stationary
```

In [57]:
```
df['Seasonal First Difference'] = df['Milk First Difference'] - df['Milk Fir
```

In [58]:
```
df['Seasonal First Difference'].plot()
```

Out[58]: <matplotlib.axes._subplots.AxesSubplot at 0x7f299e3ccd68>




In [59]:
```
df['Seasonal First Difference']
```

Out[59]:
```
Month
1962-01-01     NaN
1962-02-01     NaN
1962-03-01     NaN
1962-04-01     NaN
1962-05-01     NaN
1962-06-01     NaN
1962-07-01     NaN
1962-08-01     NaN
1962-09-01     NaN
1962-10-01     NaN
1962-11-01     NaN
1962-12-01     NaN
1963-01-01     NaN
1963-02-01    -6.0
1963-03-01     8.0
1963-04-01     4.0
1963-05-01    -2.0
1963-06-01     4.0
1963-07-01     1.0
1963-08-01    -2.0
1963-09-01    -3.0
1963-10-01    -5.0
1963-11-01     2.0
1963-12-01     4.0
1964-01-01    12.0
1964-02-01    24.0
1964-03-01   -17.0
1964-04-01    -3.0
1964-05-01    -4.0
1964-06-01    -8.0
                ...
1973-07-01    -2.0
1973-08-01    -5.0
1973-09-01    -7.0
1973-10-01     6.0
1973-11-01    13.0
1973-12-01     3.0
1974-01-01    10.0
1974-02-01    -2.0
1974-03-01     1.0
1974-04-01    -2.0
1974-05-01     8.0
1974-06-01    11.0
1974-07-01     4.0
1974-08-01     3.0
1974-09-01     1.0
1974-10-01   -10.0
1974-11-01    -8.0
1974-12-01    -2.0
1975-01-01    -5.0
1975-02-01    -2.0
1975-03-01    -1.0
1975-04-01    -2.0
1975-05-01    -4.0
1975-06-01    -7.0
```

```
1975-07-01    -2.0
1975-08-01     3.0
1975-09-01    11.0
1975-10-01    13.0
1975-11-01     9.0
1975-12-01     6.0
Name: Seasonal First Difference, Length: 168, dtype: float64
```

In [60]:
```
adf_check(df['Seasonal First Difference'].dropna())
```

```
Augmented Dickey-Fuller Test:
ADF Test Statistic : -5.038002274921979
p-value : 1.865423431878876e-05
#Lags Used : 11
Number of Observations Used : 143
strong evidence against the null hypothesis, reject the null hypothesis. Data
has no unit root and is stationary
```

In [61]:
```
df['Seasonal First Difference']
```

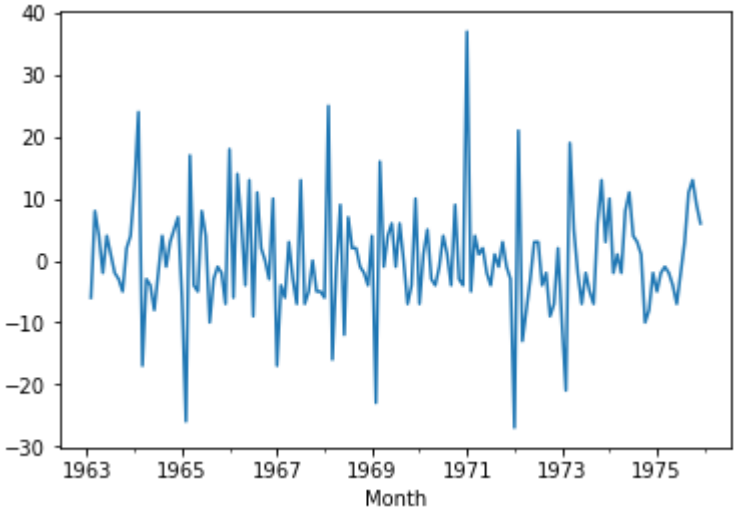
Out[61]: Month
1962-01-01 NaN
1962-02-01 NaN
1962-03-01 NaN
1962-04-01 NaN
1962-05-01 NaN
1962-06-01 NaN
1962-07-01 NaN
1962-08-01 NaN
1962-09-01 NaN
1962-10-01 NaN
1962-11-01 NaN
1962-12-01 NaN
1963-01-01 NaN
1963-02-01 -6.0
1963-03-01 8.0
1963-04-01 4.0
1963-05-01 -2.0
1963-06-01 4.0
1963-07-01 1.0
1963-08-01 -2.0
1963-09-01 -3.0
1963-10-01 -5.0
1963-11-01 2.0
1963-12-01 4.0
1964-01-01 12.0
1964-02-01 24.0
1964-03-01 -17.0
1964-04-01 -3.0
1964-05-01 -4.0
1964-06-01 -8.0
...
1973-07-01 -2.0
1973-08-01 -5.0
1973-09-01 -7.0
1973-10-01 6.0
1973-11-01 13.0
1973-12-01 3.0
1974-01-01 10.0
1974-02-01 -2.0
1974-03-01 1.0
1974-04-01 -2.0
1974-05-01 8.0
1974-06-01 11.0
1974-07-01 4.0
1974-08-01 3.0
1974-09-01 1.0
1974-10-01 -10.0
1974-11-01 -8.0
1974-12-01 -2.0
1975-01-01 -5.0
1975-02-01 -2.0
1975-03-01 -1.0
1975-04-01 -2.0
1975-05-01 -4.0
1975-06-01 -7.0

```
1975-07-01    -2.0
1975-08-01     3.0
1975-09-01    11.0
1975-10-01    13.0
1975-11-01     9.0
1975-12-01     6.0
Name: Seasonal First Difference, Length: 168, dtype: float64
```

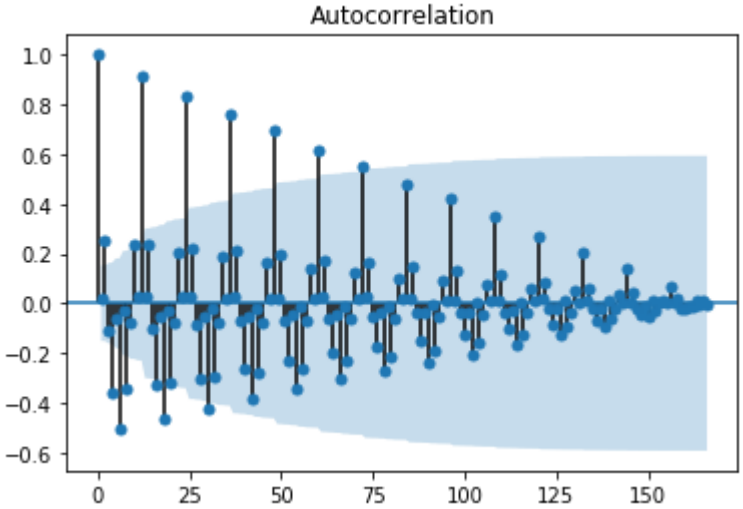
In [62]:
```
from statsmodels.graphics.tsaplots import plot_acf,plot_pacf
```

In [63]:
```
df['Seasonal First Difference'].plot()
```

Out[63]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f299e344a90>
```




In [64]:
```
fig_first = plot_acf(df["Milk First Difference"].dropna())
```




In [66]:
```
k=df["Milk First Difference"].dropna()
k
```

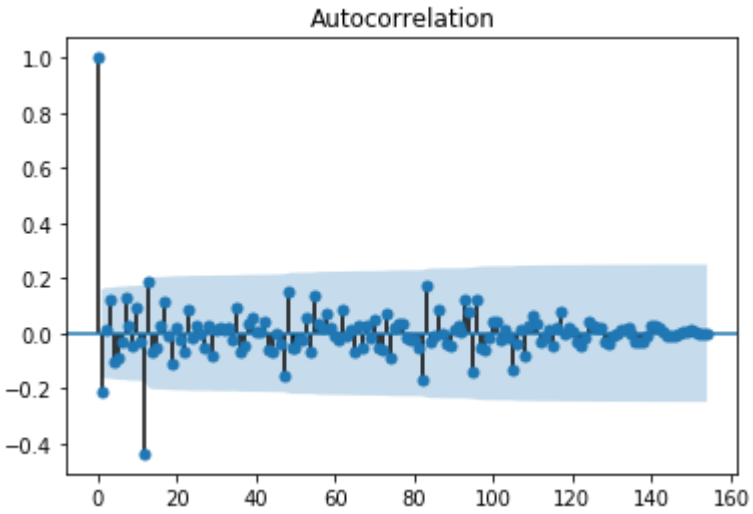
Out[66]:
```
Month
1962-02-01    -28.0
1962-03-01     79.0
1962-04-01     16.0
1962-05-01     71.0
1962-06-01    -30.0
1962-07-01    -57.0
1962-08-01    -41.0
1962-09-01    -31.0
1962-10-01      9.0
1962-11-01    -24.0
1962-12-01     29.0
1963-01-01     18.0
1963-02-01    -34.0
1963-03-01     87.0
1963-04-01     20.0
1963-05-01     69.0
1963-06-01    -26.0
1963-07-01    -56.0
1963-08-01    -43.0
1963-09-01    -34.0
1963-10-01      4.0
1963-11-01    -22.0
1963-12-01     33.0
1964-01-01     30.0
1964-02-01    -10.0
1964-03-01     70.0
1964-04-01     17.0
1964-05-01     65.0
1964-06-01    -34.0
1964-07-01    -58.0
              ...
1973-07-01    -43.0
1973-08-01    -44.0
1973-09-01    -53.0
1973-10-01      7.0
1973-11-01    -31.0
1973-12-01     42.0
1974-01-01     26.0
1974-02-01    -50.0
1974-03-01    111.0
1974-04-01     13.0
1974-05-01     67.0
1974-06-01    -22.0
1974-07-01    -39.0
1974-08-01    -41.0
1974-09-01    -52.0
1974-10-01     -3.0
1974-11-01    -39.0
1974-12-01     40.0
1975-01-01     21.0
1975-02-01    -52.0
1975-03-01    110.0
1975-04-01     11.0
1975-05-01     63.0
```



```
-29.0
```

```
1975-07-01    -41.0
1975-08-01    -38.0
1975-09-01    -41.0
1975-10-01     10.0
1975-11-01    -30.0
1975-12-01     46.0
Name: Milk First Difference, Length: 167, dtype: float64
```

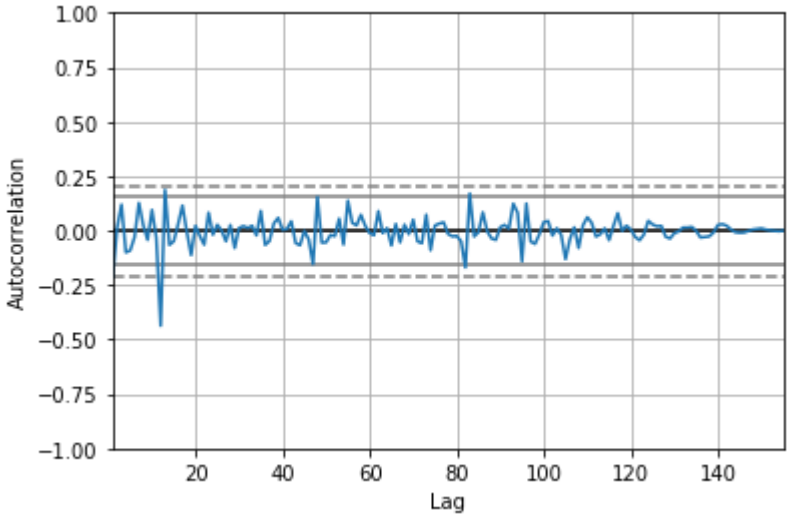
In [67]:
```
fig_seasonal_first = plot_acf(df["Seasonal First Difference"].dropna())
```



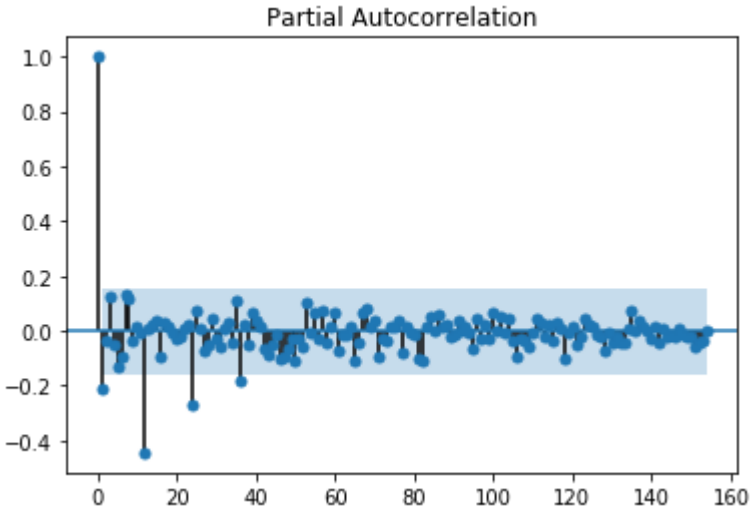

In [68]:
```
from pandas.plotting import autocorrelation_plot
```

In [69]:
```
autocorrelation_plot(df['Seasonal First Difference'].dropna())
```

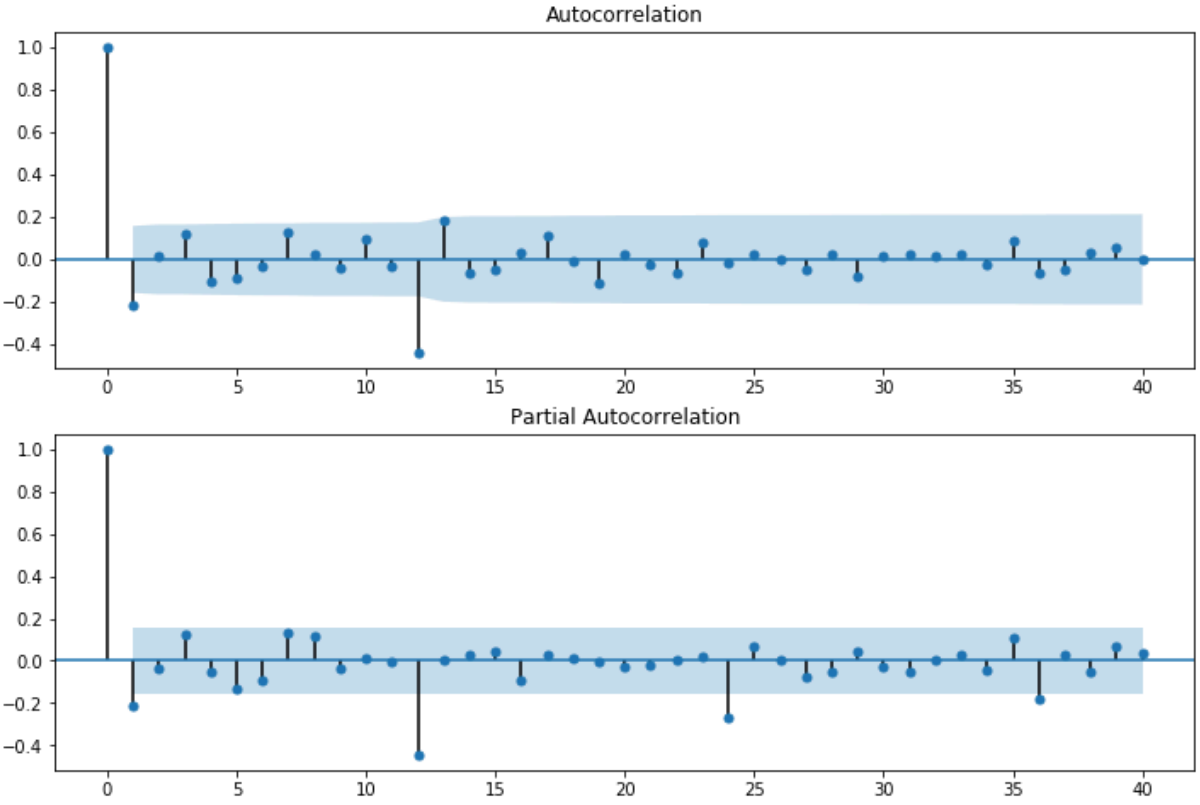
Out[69]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f299e16f278>
```




In [70]:
```
result = plot_pacf(df["Seasonal First Difference"].dropna())
```




In [71]:
```
fig = plt.figure(figsize = (12,8))
ax1 = fig.add_subplot(211)
fig = sm.graphics.tsa.plot_acf(df['Seasonal First Difference'].iloc[13:],lag
ax2 = fig.add_subplot(212)
fig = sm.graphics.tsa.plot_pacf(df['Seasonal First Difference'].iloc[13:],la
```




In [72]:
```
from statsmodels.tsa.arima_model import ARIMA
```

In [73]:
```
model = sm.tsa.statespace.SARIMAX(df['Milk in pounds per cow'],order = (0,1,
```

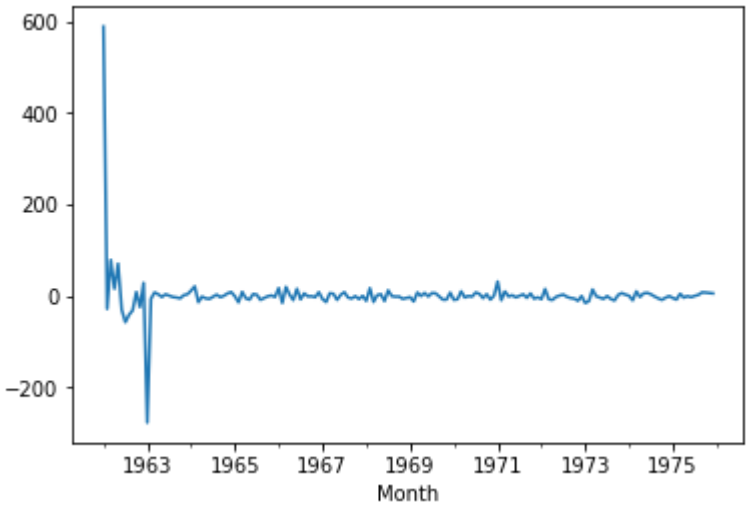
In [74]:
```
results = model.fit()
print(results.summary())
```

```
                                Statespace Model Results
==========================================================================================
Dep. Variable:             Milk in pounds per cow   No. Observations:
168
Model:             SARIMAX(0, 1, 0)x(1, 1, 1, 12)   Log Likelihood
-534.065
Date:                            Wed, 14 Aug 2019   AIC
1074.131
Time:                                    22:54:41   BIC
1083.503
Sample:                                01-01-1962   HQIC
1077.934
                                     - 12-01-1975
Covariance Type:                              opg
==============================================================================
=
                 coef    std err          z      P>|z|      [0.025      0.97
5]
------------------------------------------------------------------------------
-
ar.S.L12      -0.0449      0.106     -0.422      0.673      -0.253       0.16
3
ma.S.L12      -0.5860      0.102     -5.761      0.000      -0.785      -0.38
7
sigma2        55.5118      5.356     10.365      0.000      45.015      66.00
9
==========================================================================================
======
Ljung-Box (Q):                       33.48   Jarque-Bera (JB):
32.04
Prob(Q):                              0.76   Prob(JB):
0.00
Heteroskedasticity (H):               0.69   Skew:
0.77
Prob(H) (two-sided):                  0.18   Kurtosis:
4.60
==========================================================================================
======

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (comple
x-step).
```

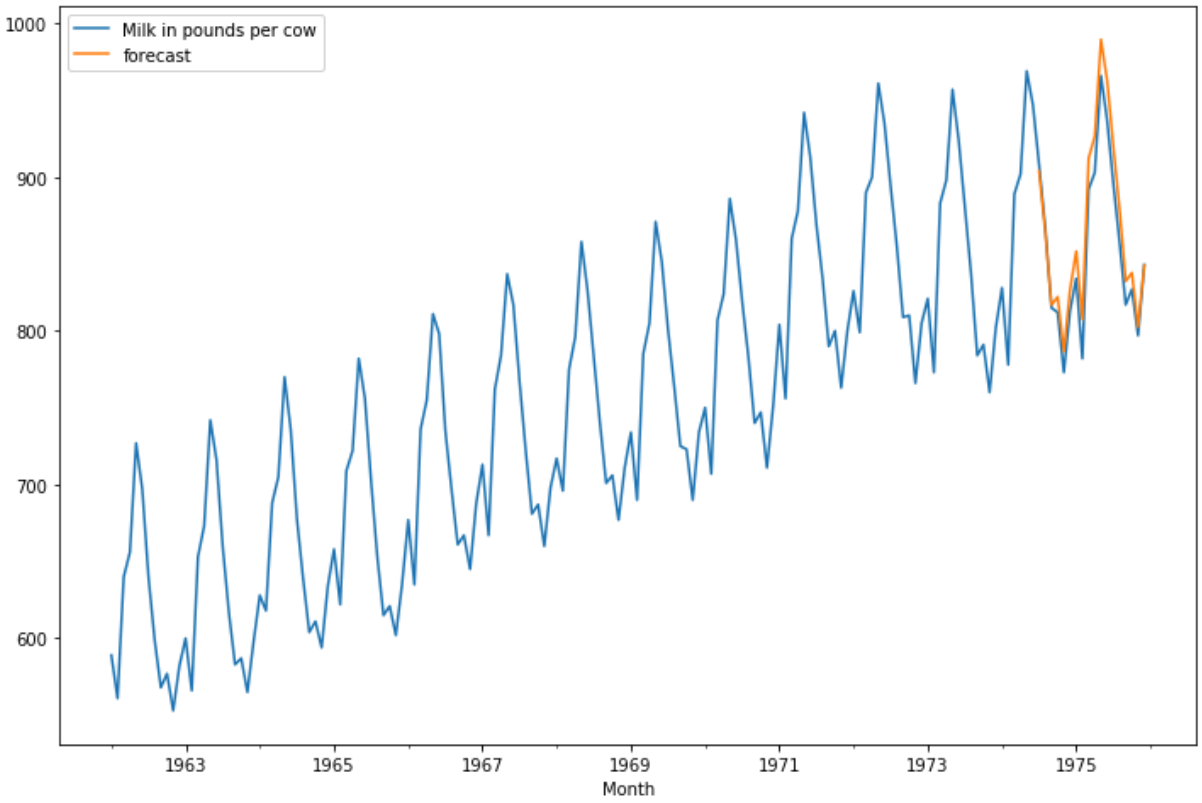
In [75]:
```
results.resid.plot()
```

Out[75]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f299dffa550>
```




In [76]:
```
df['forecast'] = results.predict(start = 150,end = 168,dynamic = True)
df[['Milk in pounds per cow','forecast']].plot(figsize = (12, 8))
```

Out[76]: <matplotlib.axes._subplots.AxesSubplot at 0x7f299df51cc0>




In [77]:
```
df.tail()
```

Out[77]:

| Month | Milk in pounds per cow | Milk First Difference | Seasonal Difference | Seasonal First Difference | forecast |
|---|---|---|---|---|---|
| 1975-08-01 | 858.0 | -38.0 | -9.0 | 3.0 | 879.668789 |
| 1975-09-01 | 817.0 | -41.0 | 2.0 | 11.0 | 832.328246 |
| 1975-10-01 | 827.0 | 10.0 | 15.0 | 13.0 | 837.721944 |
| 1975-11-01 | 797.0 | -30.0 | 24.0 | 9.0 | 802.452363 |
| 1975-12-01 | 843.0 | 46.0 | 30.0 | 6.0 | 842.499523 |

In [78]:
```
from pandas.tseries.offsets import DateOffset
```

In [79]:
```
future_dates = [df.index[-1] + DateOffset(months = x) for x in range(0,24) ]
```

In [80]:
```
future_dates
```

Out[80]:
```
[Timestamp('1975-12-01 00:00:00'),
 Timestamp('1976-01-01 00:00:00'),
 Timestamp('1976-02-01 00:00:00'),
 Timestamp('1976-03-01 00:00:00'),
 Timestamp('1976-04-01 00:00:00'),
 Timestamp('1976-05-01 00:00:00'),
 Timestamp('1976-06-01 00:00:00'),
 Timestamp('1976-07-01 00:00:00'),
 Timestamp('1976-08-01 00:00:00'),
 Timestamp('1976-09-01 00:00:00'),
 Timestamp('1976-10-01 00:00:00'),
 Timestamp('1976-11-01 00:00:00'),
 Timestamp('1976-12-01 00:00:00'),
 Timestamp('1977-01-01 00:00:00'),
 Timestamp('1977-02-01 00:00:00'),
 Timestamp('1977-03-01 00:00:00'),
 Timestamp('1977-04-01 00:00:00'),
 Timestamp('1977-05-01 00:00:00'),
 Timestamp('1977-06-01 00:00:00'),
 Timestamp('1977-07-01 00:00:00'),
 Timestamp('1977-08-01 00:00:00'),
 Timestamp('1977-09-01 00:00:00'),
 Timestamp('1977-10-01 00:00:00'),
 Timestamp('1977-11-01 00:00:00')]
```

In [81]:
```
future_dates_df = pd.DataFrame(index = future_dates[1:],columns = df.columns
```

In [82]:
```
future_df = pd.concat([df,future_dates_df])
```

In [83]:
```
future_df.head()
```

Out[83]:

| | Milk in pounds per cow | Milk First Difference | Seasonal Difference | Seasonal First Difference | forecast |
|---|---|---|---|---|---|
| **1962-01-01** | 589.0 | NaN | NaN | NaN | NaN |
| **1962-02-01** | 561.0 | -28.0 | NaN | NaN | NaN |
| **1962-03-01** | 640.0 | 79.0 | NaN | NaN | NaN |
| **1962-04-01** | 656.0 | 16.0 | NaN | NaN | NaN |
| **1962-05-01** | 727.0 | 71.0 | NaN | NaN | NaN |

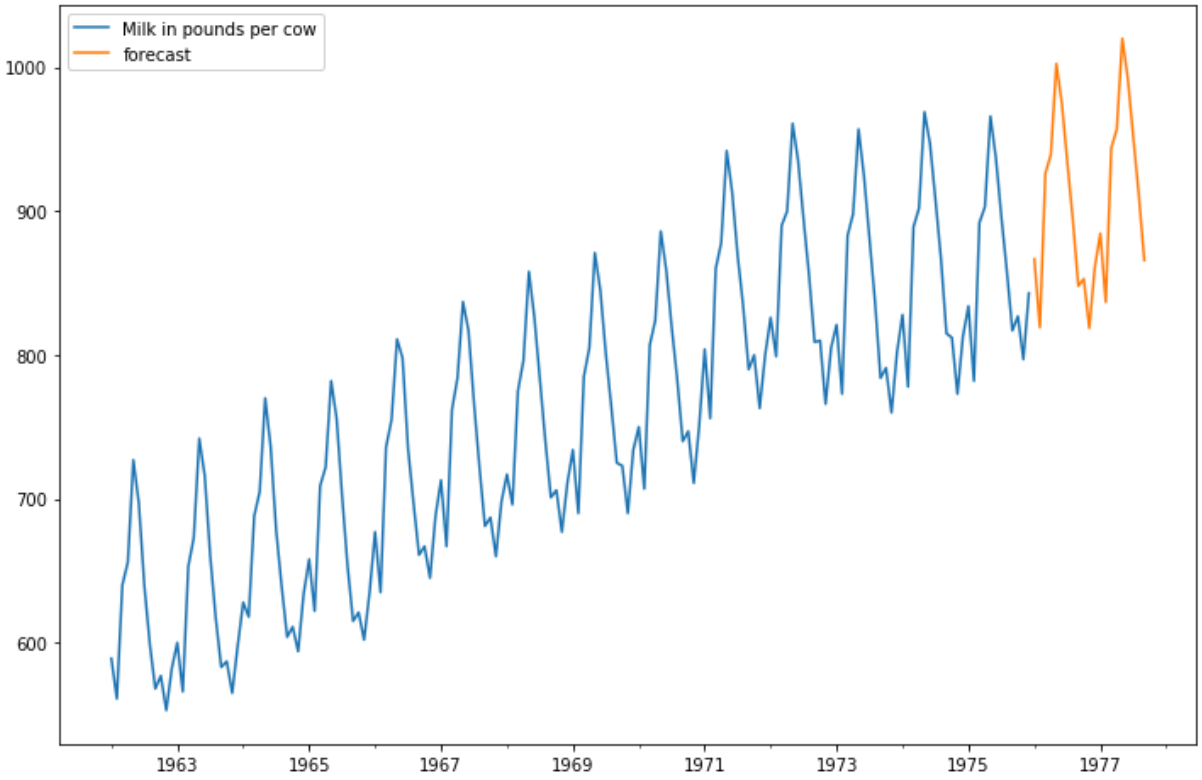
In [84]:
```
future_df.tail()
```

Out[84]:

| | Milk in pounds per cow | Milk First Difference | Seasonal Difference | Seasonal First Difference | forecast |
|---|---|---|---|---|---|
| **1977-07-01** | NaN | NaN | NaN | NaN | NaN |
| **1977-08-01** | NaN | NaN | NaN | NaN | NaN |
| **1977-09-01** | NaN | NaN | NaN | NaN | NaN |
| **1977-10-01** | NaN | NaN | NaN | NaN | NaN |
| **1977-11-01** | NaN | NaN | NaN | NaN | NaN |

In [85]:
```
future_df['forecast'] = results.predict(start = 168, end = 188,dynamic= True
future_df[['Milk in pounds per cow', 'forecast']].plot(figsize = (12, 8))
```

Out[85]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2987fb3828>
```




In [ ]: