In [83]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.arima_model import ARMA
%matplotlib inline
```

In [84]:
```
df=pd.read_csv('/Users/kushimahar/Downloads/cloth_sales.csv')
df.head()
```

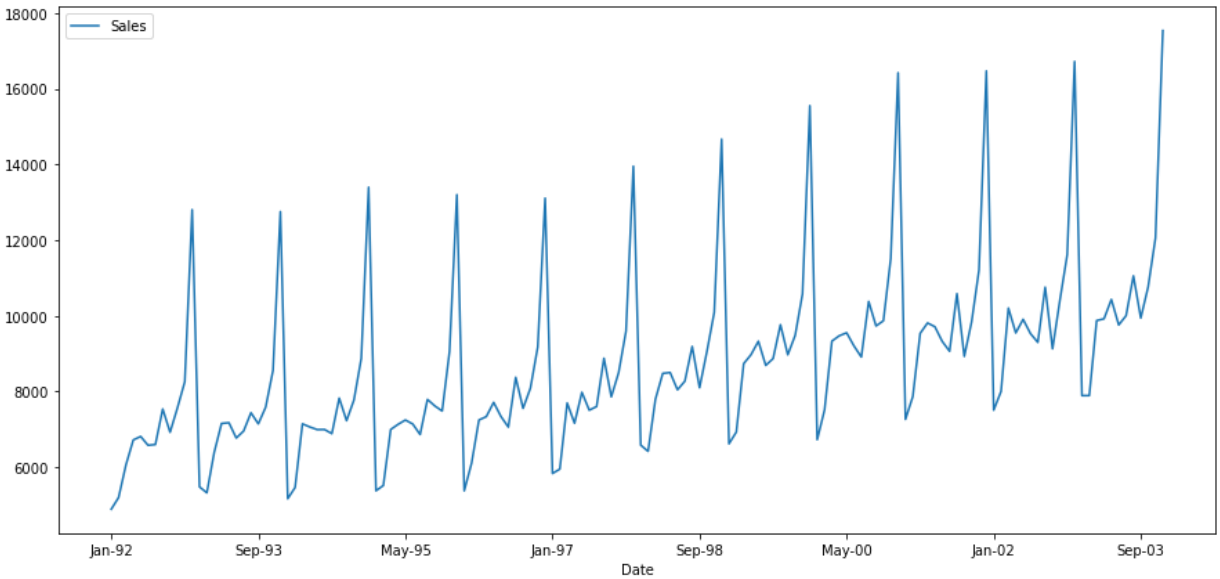
Out[84]:

	date	sales
0	Jan-92	4889
1	Feb-92	5198
2	Mar-92	6061
3	Apr-92	6720
4	May-92	6811

In [85]:
```
df.columns=["Date","Sales"]
df.head()
df.describe()
df.set_index('Date',inplace=True)
```

In [86]:
```
from pylab import rcParams
rcParams['figure.figsize'] = 15, 7
df.plot()
```

Out[86]:
```
<AxesSubplot:xlabel='Date'>
```



In [87]:
```
from statsmodels.tsa.stattools import adfuller
```

In [88]:
```
test_result=adfuller(df['Sales'])
```

In [89]:
```
#Ho: It is non-stationary
```

```
#H1: It is stationary
```

In [90]:

```
def adfuller_test(sales):
    result=adfuller(sales)
    labels = ['ADF Test Statistic','p-value','#Lags Used','Number of Observat
    for value,label in zip(result,labels):
        print(label+' : '+str(value) )

    if result[1] <= 0.05:
        print("strong evidence against the null hypothesis(Ho), reject the nu
    else:
        print("weak evidence against null hypothesis,indicating it is non-sta

adfuller_test(df['Sales'])
```

```
ADF Test Statistic : 0.6384826676448148
p-value : 0.9885267347007706
#Lags Used : 14
Number of Observations : 129
weak evidence against null hypothesis,indicating it is non-stationary
```

In [91]:

```
df['Sales First Difference'] = df['Sales'] - df['Sales'].shift(1)
df['Seasonal First Difference']=df['Sales']-df['Sales'].shift(12)
df.head()
```

Out[91]:

Date	Sales	Sales First Difference	Seasonal First Difference
Jan-92	4889	NaN	NaN
Feb-92	5198	309.0	NaN
Mar-92	6061	863.0	NaN
Apr-92	6720	659.0	NaN
May-92	6811	91.0	NaN

In [92]:

```
# Again testing if data is stationary
adfuller_test(df['Seasonal First Difference'].dropna())
```

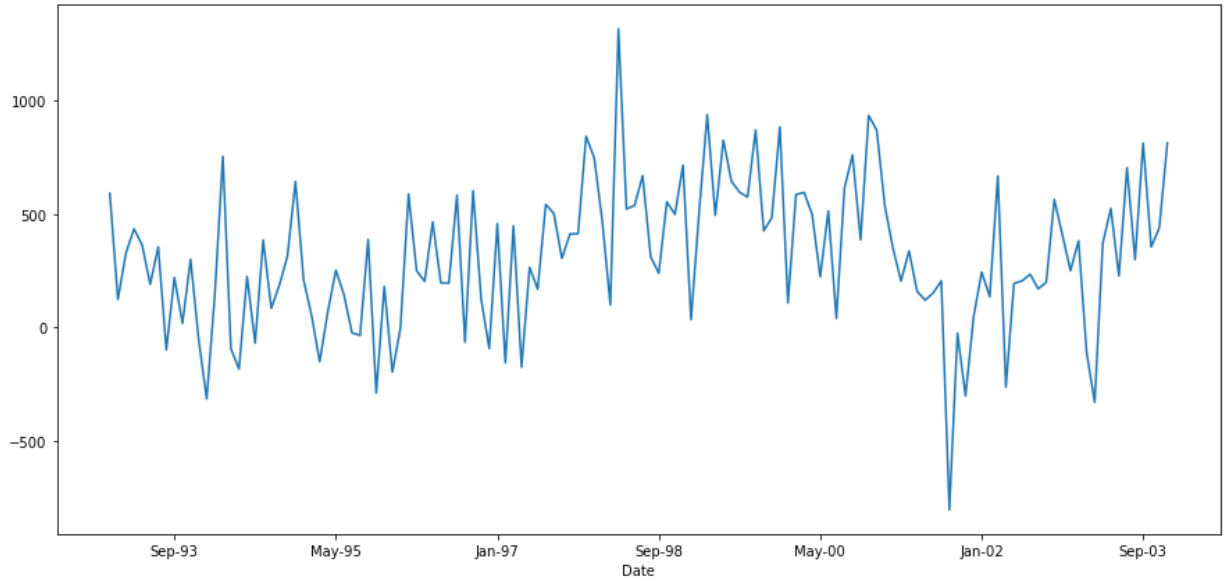
```
ADF Test Statistic : -3.442558535470895
p-value : 0.009590834550182063
#Lags Used : 2
Number of Observations : 129
strong evidence against the null hypothesis(Ho), reject the null hypothesis. D
ata is stationary
```

In [93]:

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

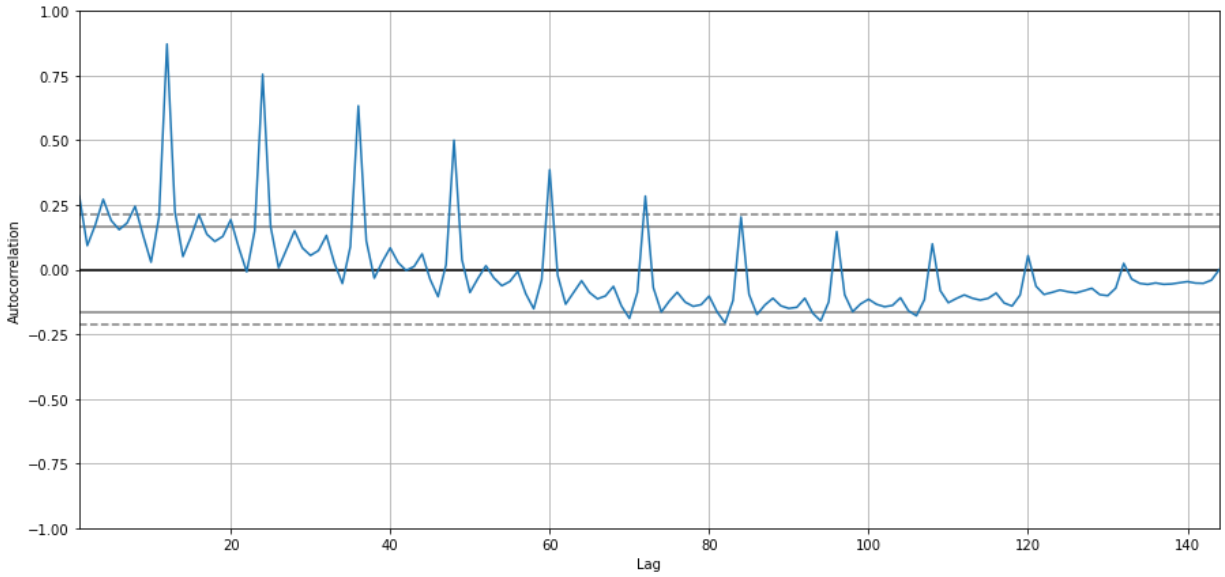
Out[93]:

```
<AxesSubplot:xlabel='Date'>
```



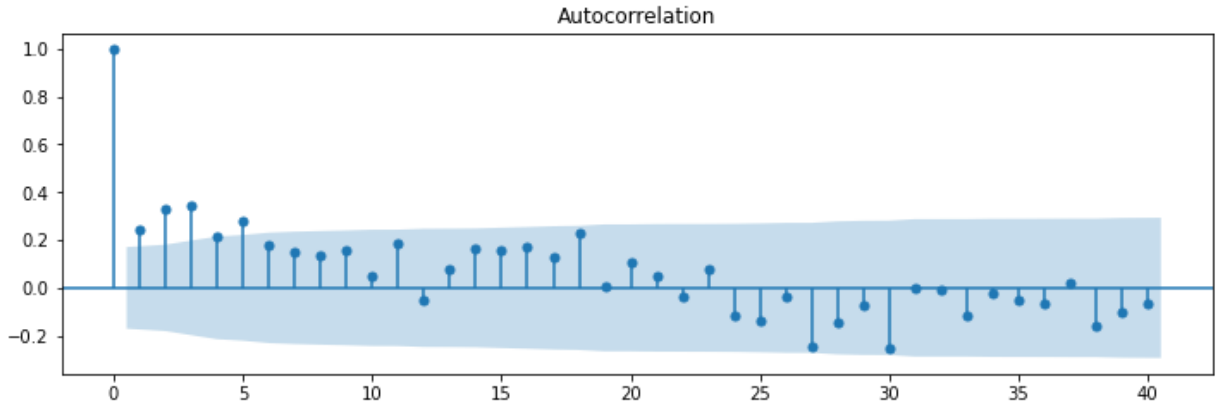
In [94]:

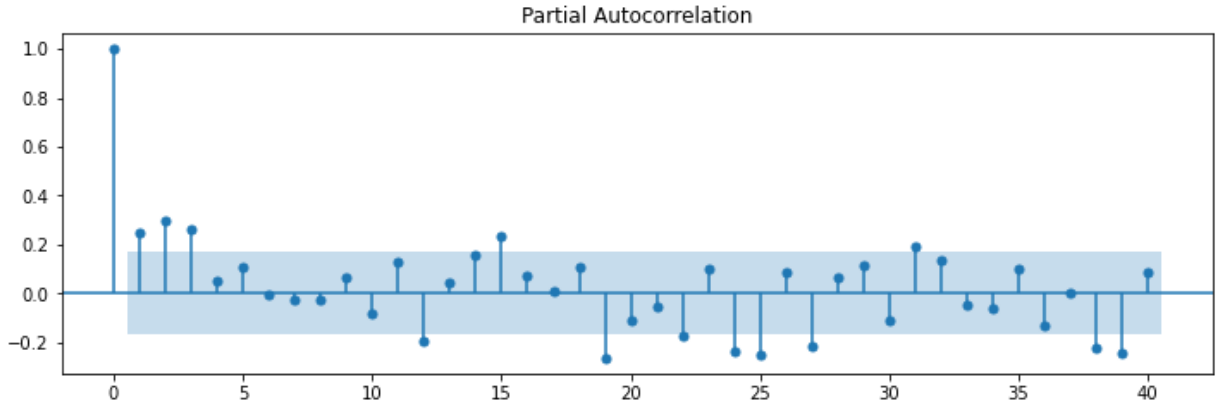
```
from pandas.plotting import autocorrelation_plot
autocorrelation_plot(df['Sales'])
plt.show()
```



In [95]:

```
from statsmodels.graphics.tsaplots import plot_acf,plot_pacf
import statsmodels.api as sm
fig = plt.figure(figsize=(12,8))
ax1 = fig.add_subplot(211)
fig = sm.graphics.tsa.plot_acf(df['Seasonal First Difference'].dropna(),lags=
ax2 = fig.add_subplot(212)
fig = sm.graphics.tsa.plot_pacf(df['Seasonal First Difference'].dropna(),lags
```





In [96]:

```
model = ARMA (df['Sales'], order = (3,0))
res = model.fit()
res.plot_predict()
model_fit=model.fit()
model_fit.summary()
```

```
/Users/kushimahar/opt/anaconda3/lib/python3.8/site-packages/statsmodels/tsa/ar
ima_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)
/Users/kushimahar/opt/anaconda3/lib/python3.8/site-packages/statsmodels/tsa/ba
se/tsa_model.py:578: ValueWarning: An unsupported index was provided and will
be ignored when e.g. forecasting.
  warnings.warn('An unsupported index was provided and will be'
```

Out[96]:

ARMA Model Results

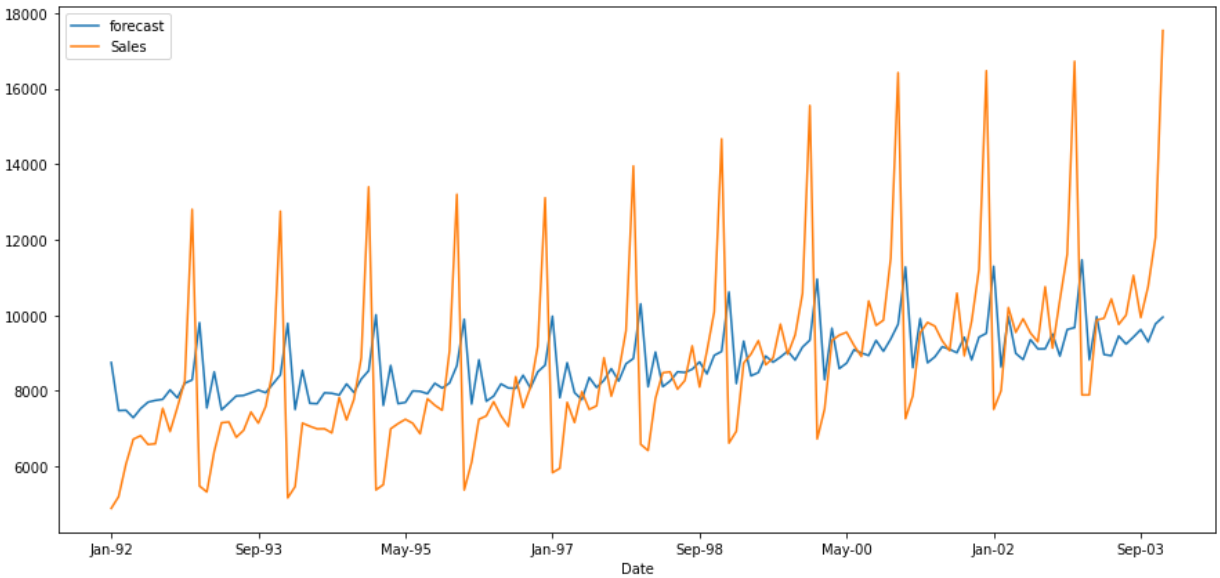
Dep. Variable:	Sales	**No. Observations:**	144
Model:	ARMA(3, 0)	**Log Likelihood**	-1315.322
Method:	css-mle	**S.D. of innovations**	2240.561
Date:	Fri, 29 Apr 2022	**AIC**	2640.644

Time:	10:20:52	**BIC**	2655.493
Sample:	0	**HQIC**	2646.678

	coef	std err	z	P>|z|	[0.025	0.975]
const	8747.5907	349.534	25.026	0.000	8062.516	9432.665
ar.L1.Sales	0.3112	0.086	3.632	0.000	0.143	0.479
ar.L2.Sales	-0.0233	0.089	-0.261	0.794	-0.199	0.152
ar.L3.Sales	0.1831	0.087	2.112	0.035	0.013	0.353

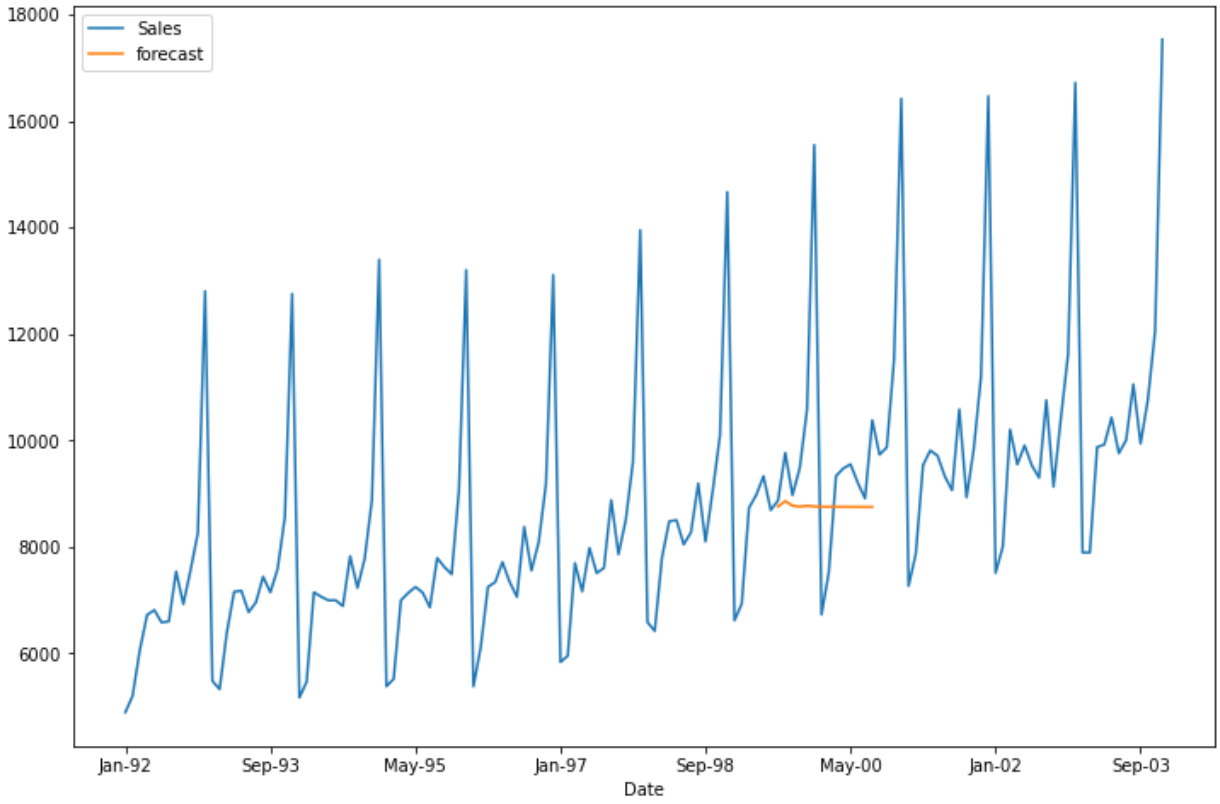
Roots

	Real	Imaginary	Modulus	Frequency
AR.1	1.4778	-0.0000j	1.4778	-0.0000
AR.2	-0.6752	-1.7998j	1.9223	-0.3071
AR.3	-0.6752	+1.7998j	1.9223	0.3071



In [97]:
```
df['forecast']=model_fit.predict(start=90,end=103,dynamic=True)
df[['Sales','forecast']].plot(figsize=(12,8))
```

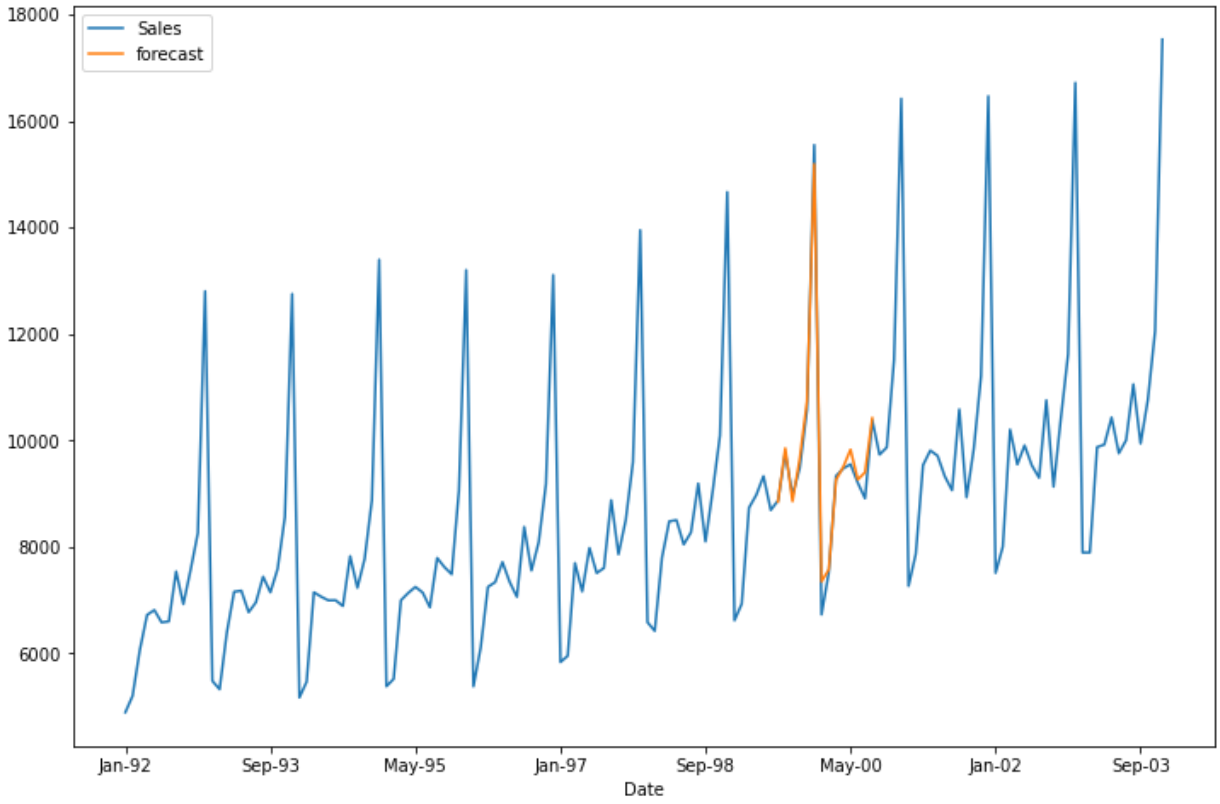
Out[97]: <AxesSubplot:xlabel='Date'>



In [98]:
```
import statsmodels.api as sm
model=sm.tsa.statespace.SARIMAX(df['Sales'],order=(1, 1, 1),seasonal_order=(1
results=model.fit()
df['forecast']=results.predict(start=90,end=103,dynamic=True)
df[['Sales','forecast']].plot(figsize=(12,8))
```

```
/Users/kushimahar/opt/anaconda3/lib/python3.8/site-packages/statsmodels/tsa/ba
se/tsa_model.py:578: ValueWarning: An unsupported index was provided and will
be ignored when e.g. forecasting.
  warnings.warn('An unsupported index was provided and will be'
/Users/kushimahar/opt/anaconda3/lib/python3.8/site-packages/statsmodels/tsa/ba
se/tsa_model.py:578: ValueWarning: An unsupported index was provided and will
be ignored when e.g. forecasting.
  warnings.warn('An unsupported index was provided and will be'
```

Out[98]: <AxesSubplot:xlabel='Date'>



In [99]:

```
# For non-seasonal data
#p=1, d=1, q=0 or 1

from statsmodels.tsa.arima_model import ARIMA
model=ARIMA(df['Sales'],order=(1,1,1))
model_fit=model.fit()
model_fit.summary()
```

```
/Users/kushimahar/opt/anaconda3/lib/python3.8/site-packages/statsmodels/tsa/ar
ima_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)
/Users/kushimahar/opt/anaconda3/lib/python3.8/site-packages/statsmodels/tsa/ba
se/tsa_model.py:578: ValueWarning: An unsupported index was provided and will
be ignored when e.g. forecasting.
  warnings.warn('An unsupported index was provided and will be'
/Users/kushimahar/opt/anaconda3/lib/python3.8/site-packages/statsmodels/tsa/ba
se/tsa_model.py:578: ValueWarning: An unsupported index was provided and will
be ignored when e.g. forecasting.
  warnings.warn('An unsupported index was provided and will be'
/Users/kushimahar/opt/anaconda3/lib/python3.8/site-packages/statsmodels/tsa/ar
ima_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)
```

Out[99]:

ARIMA Model Results

Dep. Variable:	D.Sales	**No. Observations:**	143
Model:	ARIMA(1, 1, 1)	**Log Likelihood**	-1293.257
Method:	css-mle	**S.D. of innovations**	2013.668
Date:	Fri, 29 Apr 2022	**AIC**	2594.514
Time:	10:20:55	**BIC**	2606.366
Sample:	1	**HQIC**	2599.330

	coef	std err	z	P>|z|	[0.025	0.975]
const	31.1829	4.198	7.428	0.000	22.955	39.411
ar.L1.D.Sales	0.0378	0.087	0.433	0.665	-0.133	0.209
ma.L1.D.Sales	-1.0000	0.019	-53.202	0.000	-1.037	-0.963

Roots

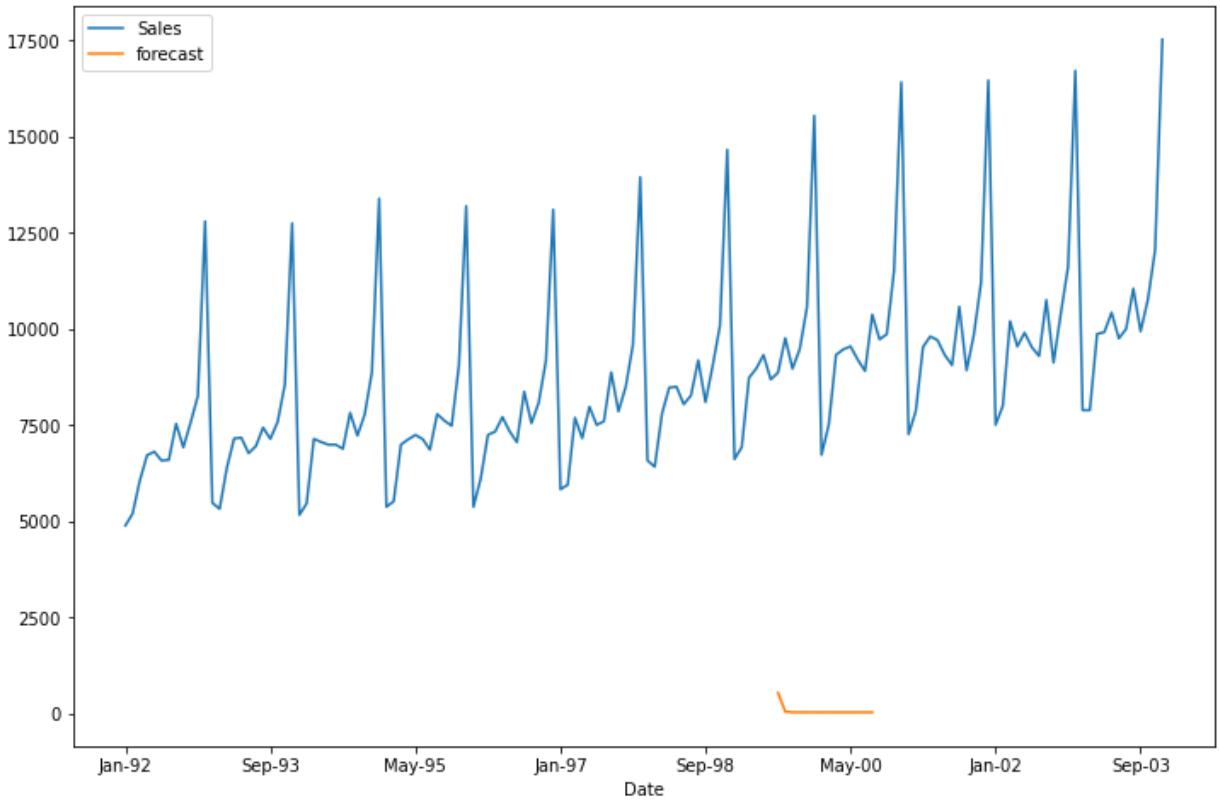
	Real	Imaginary	Modulus	Frequency
AR.1	26.4512	+0.0000j	26.4512	0.0000
MA.1	1.0000	+0.0000j	1.0000	0.0000

In [100…

```
df['forecast']=model_fit.predict(start=90,end=103,dynamic=True)
df[['Sales','forecast']].plot(figsize=(12,8))
```

Out[100…

```
<AxesSubplot:xlabel='Date'>
```

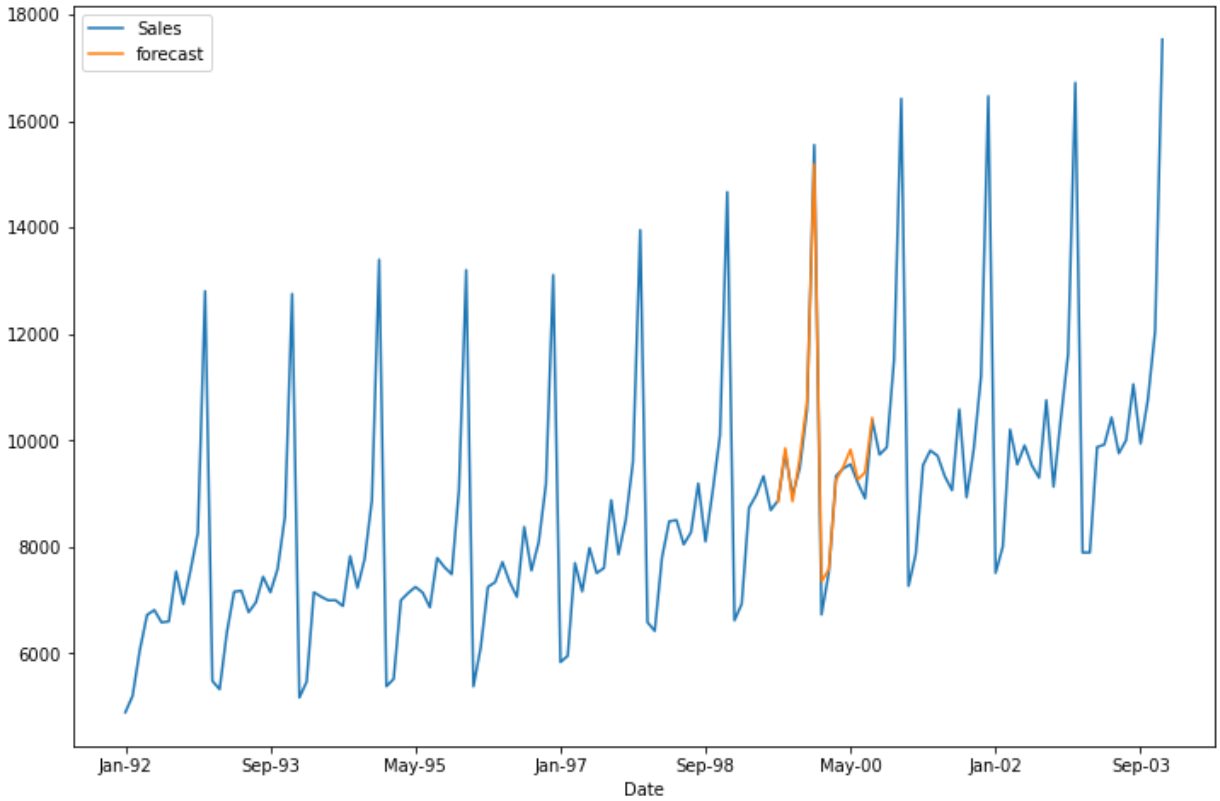


In [101…

```
import statsmodels.api as sm
model=sm.tsa.statespace.SARIMAX(df['Sales'],order=(1, 1, 1),seasonal_order=(1
results=model.fit()
df['forecast']=results.predict(start=90,end=103,dynamic=True)
df[['Sales','forecast']].plot(figsize=(12,8))
```

```
/Users/kushimahar/opt/anaconda3/lib/python3.8/site-packages/statsmodels/tsa/ba
se/tsa_model.py:578: ValueWarning: An unsupported index was provided and will
be ignored when e.g. forecasting.
  warnings.warn('An unsupported index was provided and will be'
/Users/kushimahar/opt/anaconda3/lib/python3.8/site-packages/statsmodels/tsa/ba
se/tsa_model.py:578: ValueWarning: An unsupported index was provided and will
be ignored when e.g. forecasting.
  warnings.warn('An unsupported index was provided and will be'
```

Out[101… <AxesSubplot:xlabel='Date'>


Sales
forecast
18000
16000
14000
12000
10000
8000
6000
Jan-92
Sep-93
May-95
Jan-97
Sep-98
May-00
Jan-02
Sep-03
Date

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: