In [144…
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

In [145…
```
df = pd.read_csv('Problem1_DataSet.csv')
df.head()
```

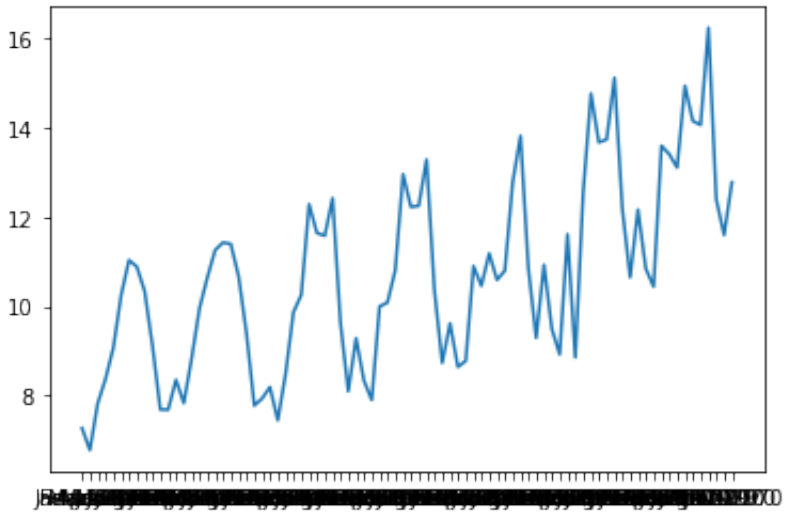
Out[145]:

	Month	Miles, in Millions
0	Jan-1964	7.269
1	Feb-1964	6.775
2	Mar-1964	7.819
3	Apr-1964	8.371
4	May-1964	9.069

Q1 a)

In [146…
```
x = df['Month']
y = df['Miles, in Millions']
plt.plot(x,y)
```

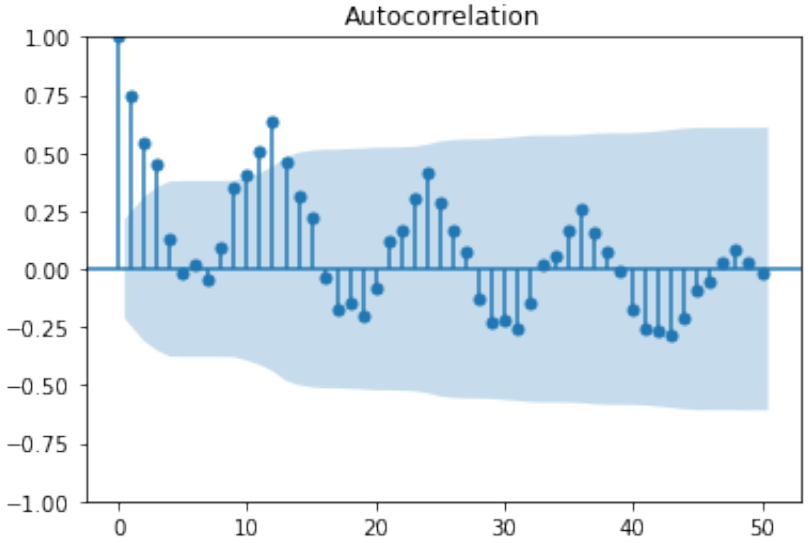
Out[146]:
```
[<matplotlib.lines.Line2D at 0x7fb9b6a776d0>]
```



Q1 b)

In [147…
```
import statsmodels.api as sm
```

In [148…
```
sm.graphics.tsa.plot_acf(df['Miles, in Millions'], lags=50);
# 11 period
```



In [149…
```
df['3day_ma'] = df.rolling(window=3).mean()
# df['4day_ma'] = df.rolling(window=4).mean()
# df['5day_ma'] = df.rolling(window=5).mean()
# df['6day_ma'] = df.rolling(window=6).mean()
```

```
/var/folders/9l/t9hntp494tqc0sdmv_99jt1w0000gn/T/ipykernel_43712/3171510369.
py:1: FutureWarning: Dropping of nuisance columns in rolling operations is d
eprecated; in a future version this will raise TypeError. Select only valid
columns before calling the operation. Dropped columns were Index(['Month'],
dtype='object')
  df['3day_ma'] = df.rolling(window=3).mean()
```

In [150…
```
df['4day_ma'] = df.iloc[:,1].rolling(window=4).mean()
```

In [151…
```
df.head()
```

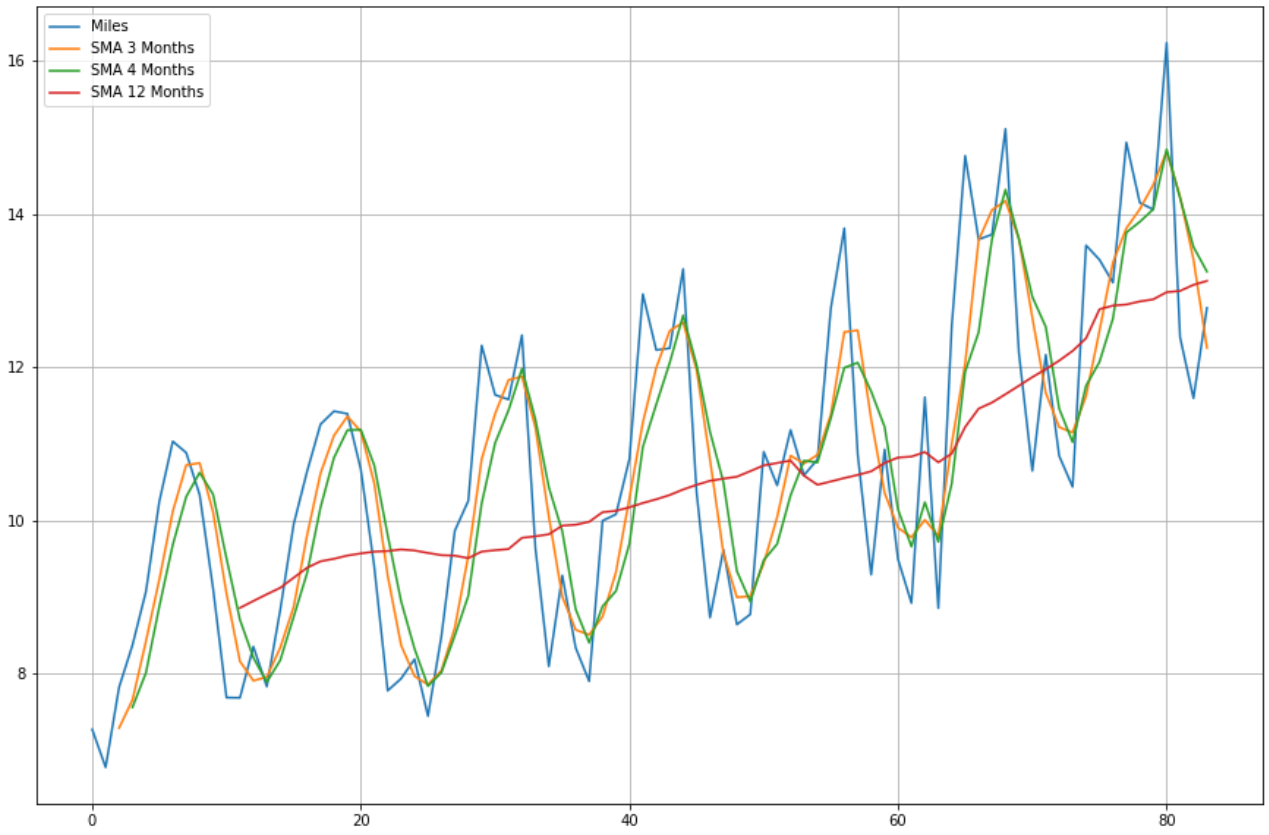
Out[151]:

	Month	Miles, in Millions	3day_ma	4day_ma
0	Jan-1964	7.269	NaN	NaN
1	Feb-1964	6.775	NaN	NaN
2	Mar-1964	7.819	7.287667	NaN
3	Apr-1964	8.371	7.655000	7.5585
4	May-1964	9.069	8.419667	8.0085

In [199…
```
df['12day_ma'] = df.iloc[:,1].rolling(window=12).mean()
```

In [200…
```
plt.figure(figsize=(15,10))
plt.grid(True)
plt.plot(df['Miles, in Millions'],label='Miles')
plt.plot(df['3day_ma'],label='SMA 3 Months')
plt.plot(df['4day_ma'],label='SMA 4 Months')
plt.plot(df['12day_ma'],label='SMA 12 Months')
plt.legend(loc=2)
```

Out[200]:
```
<matplotlib.legend.Legend at 0x7fb9b8c02a60>
```



The trend is increasing.

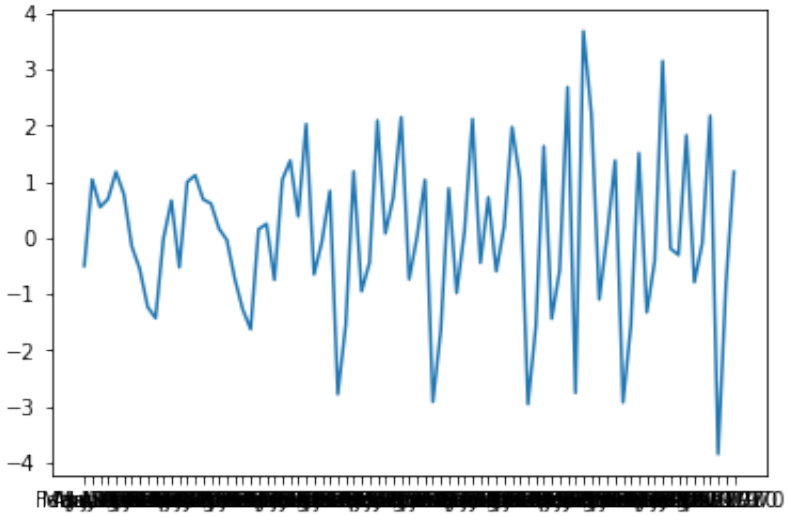
In [ ]:
```
```

In [153…
```
df2 = df[['Month', 'Miles, in Millions']].set_index(['Month'])
```

In [154…
```
from statsmodels.tsa.statespace.tools import diff
diff1 = diff(df2)
```

In [155…
```
plt.plot(diff1,label='first difference')
```

Out[155]:
```
[<matplotlib.lines.Line2D at 0x7fb9e37e7d30>]
```



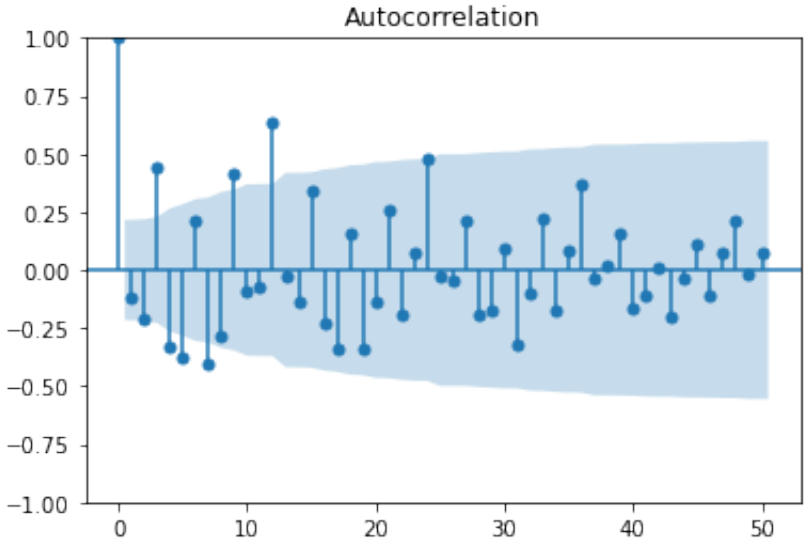
In [156…
```
diff1
```

Out[156]:

Month	Miles, in Millions
Feb-1964	-0.494
Mar-1964	1.044
Apr-1964	0.552
May-1964	0.698
Jun-1964	1.179
...	...
Aug-1970	-0.090
Sep-1970	2.177
Oct-1970	-3.845
Nov-1970	-0.795
Dec-1970	1.178

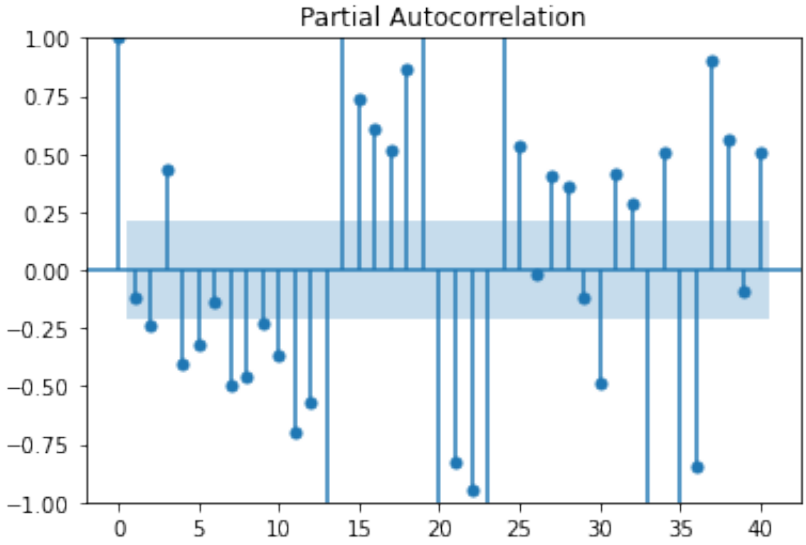
83 rows × 1 columns

In [157…
```
sm.graphics.tsa.plot_acf(diff1,lags=50);
```



In [158…
```
sm.graphics.tsa.plot_pacf(diff1,lags=40);
```

```
/Users/dhavalgarg/opt/anaconda3/lib/python3.9/site-packages/statsmodels/graphics/tsaplots.py:348: FutureWarning: The default method 'yw' can produce PACF values outside of the [-1,1] interval. After 0.13, the default will change tounadjusted Yule-Walker ('ywm'). You can use this method now by setting method='ywm'.
  warnings.warn(
```



In [161…
```
diff2 = diff1.diff(12)
```

In [162…
```
diff2
```

Out[162]:

Month	Miles, in Millions
Feb-1964	NaN
Mar-1964	NaN
Apr-1964	NaN
May-1964	NaN
Jun-1964	NaN
...	...
Aug-1970	-0.154
Sep-1970	0.798
Oct-1970	-0.920
Nov-1970	0.745
Dec-1970	-0.338

83 rows × 1 columns

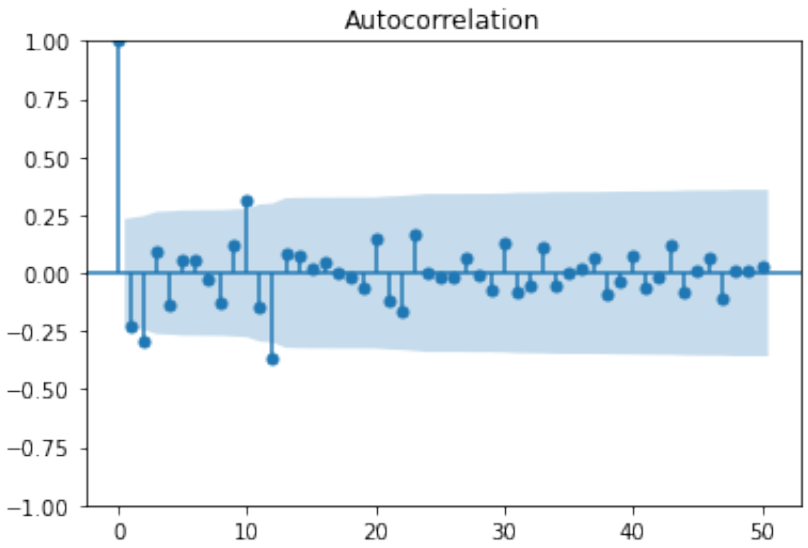
In [163…

```
diff2.head(15)
```

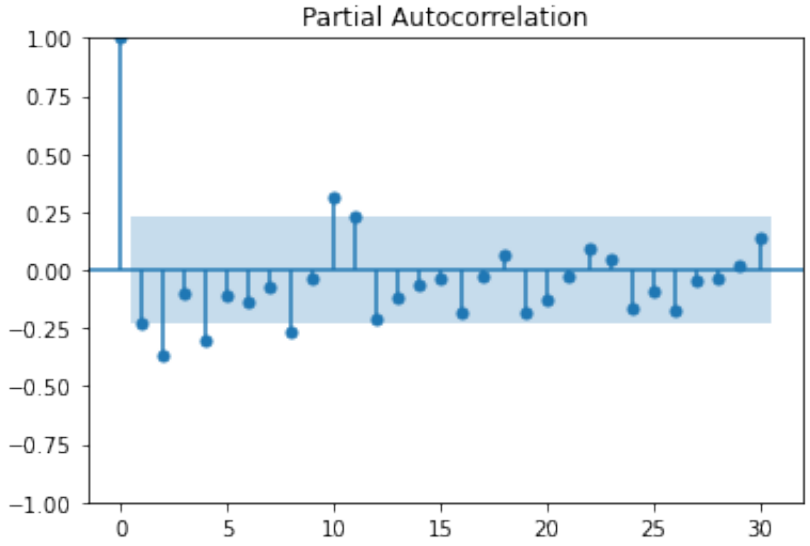
Out[163]:

Month	Miles, in Millions
Feb-1964	NaN
Mar-1964	NaN
Apr-1964	NaN
May-1964	NaN
Jun-1964	NaN
Jul-1964	NaN
Aug-1964	NaN
Sep-1964	NaN
Oct-1964	NaN
Nov-1964	NaN
Dec-1964	NaN
Jan-1965	NaN
Feb-1965	-0.027
Mar-1965	-0.044
Apr-1965	0.567

In [165…
```
sm.graphics.tsa.plot_acf(diff2[12:],lags=50);
```

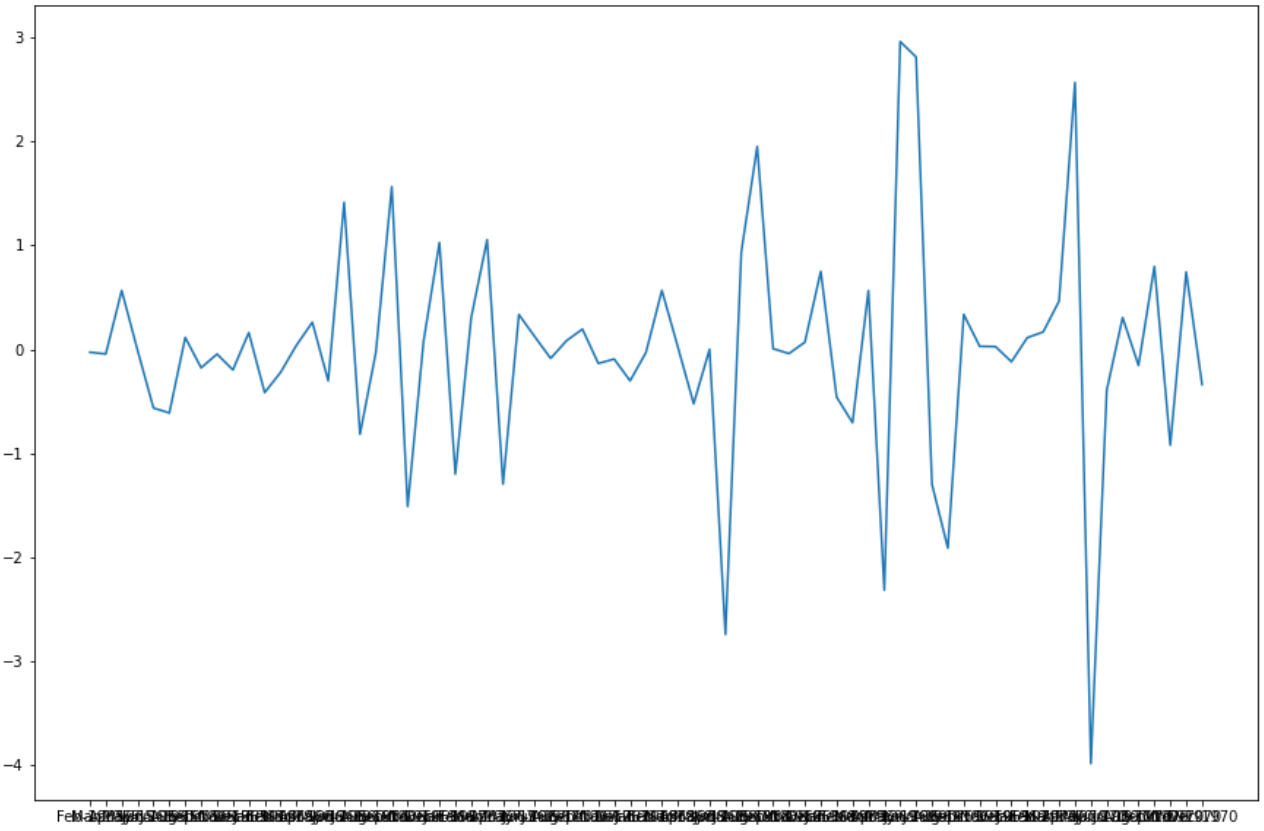


In [166…
```
sm.graphics.tsa.plot_pacf(diff2[12:],lags=30,method='ywm');
```



In [167…
```
plt.figure(figsize=(15,10))
plt.plot(diff2[12:])
```

Out[167]:
```
[<matplotlib.lines.Line2D at 0x7fb9b8277550>]
```



In [168…
```
from statsmodels.tsa.stattools import adfuller
dftest = adfuller(diff2[12:], autolag = 'AIC')
print("1. ADF : ",dftest[0])
print("2. P-Value : ", dftest[1])
print("3. Num Of Lags : ", dftest[2])
print("4. Num Of Observations Used For ADF Regression and Critical Values Ca
print("5. Critical Values :")
for key, val in dftest[4].items():
    print("\t",key, ": ", val)
```

```
1. ADF :  -2.6064342517214247
2. P-Value :  0.09167471706254021
3. Num Of Lags :  12
4. Num Of Observations Used For ADF Regression and Critical Values Calculati
on : 58
5. Critical Values :
         1% :  -3.548493559596539
         5% :  -2.912836594776334
         10% :  -2.594129155766944
```

In [177…
```
df['Date'] = pd.to_datetime(df['Month'])
df.head()
```

Out[177]:

	Month	Miles, in Millions	3day_ma	4day_ma	Date
0	Jan-1964	7.269	NaN	NaN	1964-01-01
1	Feb-1964	6.775	NaN	NaN	1964-02-01
2	Mar-1964	7.819	7.287667	NaN	1964-03-01
3	Apr-1964	8.371	7.655000	7.5585	1964-04-01
4	May-1964	9.069	8.419667	8.0085	1964-05-01

In [178…
```
training_data=df[df.Date < pd.to_datetime("1970-01-01")]
```

In [179…
```
training_data.head()
```

Out[179]:

	Month	Miles, in Millions	3day_ma	4day_ma	Date
0	Jan-1964	7.269	NaN	NaN	1964-01-01
1	Feb-1964	6.775	NaN	NaN	1964-02-01
2	Mar-1964	7.819	7.287667	NaN	1964-03-01
3	Apr-1964	8.371	7.655000	7.5585	1964-04-01
4	May-1964	9.069	8.419667	8.0085	1964-05-01

In [205…
```
test_data=df[df.Date >= pd.to_datetime("1970-01-01")]
test_data.head()
```

Out[205]:

	Month	Miles, in Millions	3day_ma	4day_ma	Date	12day_ma
72	Jan-1970	10.840	11.215333	11.45775	1970-01-01	12.084417
73	Feb-1970	10.436	11.145667	11.02050	1970-02-01	12.210833
74	Mar-1970	13.589	11.621667	11.75650	1970-03-01	12.376000
75	Apr-1970	13.402	12.475667	12.06675	1970-04-01	12.755167
76	May-1970	13.103	13.364667	12.63250	1970-05-01	12.802333

In [210…
```
from pmdarima import auto_arima
auto_arima(
    df['Miles, in Millions'],
    start_p = 1, max_p = 12,
    start_q = 1, max_q = 12,
    seasonal = True, trace = False).summary()
```

```
/Users/dhavalgarg/opt/anaconda3/lib/python3.9/site-packages/statsmodels/tsa/
statespace/sarimax.py:1899: RuntimeWarning: invalid value encountered in rec
iprocal
  return np.roots(self.polynomial_reduced_ar)**-1
/Users/dhavalgarg/opt/anaconda3/lib/python3.9/site-packages/statsmodels/tsa/
statespace/sarimax.py:1906: RuntimeWarning: invalid value encountered in rec
iprocal
  return np.roots(self.polynomial_reduced_ma)**-1
/Users/dhavalgarg/opt/anaconda3/lib/python3.9/site-packages/statsmodels/tsa/
statespace/sarimax.py:1899: RuntimeWarning: invalid value encountered in rec
iprocal
  return np.roots(self.polynomial_reduced_ar)**-1
/Users/dhavalgarg/opt/anaconda3/lib/python3.9/site-packages/statsmodels/tsa/
statespace/sarimax.py:1906: RuntimeWarning: invalid value encountered in rec
iprocal
  return np.roots(self.polynomial_reduced_ma)**-1
```

Out[210]:

SARIMAX Results

Dep. Variable:	y	**No. Observations:**	84
Model:	SARIMAX(0, 1, 0)	**Log Likelihood**	-149.034
Date:	Mon, 31 Oct 2022	**AIC**	300.068
Time:	23:06:48	**BIC**	302.486
Sample:	0	**HQIC**	301.039
	- 84		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
sigma2	2.1240	0.328	6.484	0.000	1.482	2.766

Ljung-Box (L1) (Q):	1.26	**Jarque-Bera (JB):**	0.46
Prob(Q):	0.26	**Prob(JB):**	0.79
Heteroskedasticity (H):	5.13	**Skew:**	-0.18
Prob(H) (two-sided):	0.00	**Kurtosis:**	3.06

Warnings:

[1] Covariance matrix calculated using the outer product of gradients (complex-step).

In [228…

```
np.asarray(training_data)
```

Out[228]:

```
array([['Jan-1964', 7.269, nan, nan, Timestamp('1964-01-01 00:00:00')],
       ['Feb-1964', 6.775, nan, nan, Timestamp('1964-02-01 00:00:00')],
       ['Mar-1964', 7.819, 7.2876666666666665, nan,
        Timestamp('1964-03-01 00:00:00')],
       ['Apr-1964', 8.371, 7.655000000000001, 7.5585,
        Timestamp('1964-04-01 00:00:00')],
       ['May-1964', 9.069, 8.419666666666668, 8.008500000000002,
        Timestamp('1964-05-01 00:00:00')],
       ['Jun-1964', 10.248, 9.229333333333335, 8.876750000000001,
        Timestamp('1964-06-01 00:00:00')],
       ['Jul-1964', 11.03, 10.115666666666668, 9.6795,
        Timestamp('1964-07-01 00:00:00')],
       ['Aug-1964', 10.882, 10.719999999999999, 10.30725,
        Timestamp('1964-08-01 00:00:00')],
       ['Sep-1964', 10.333, 10.748333333333335, 10.623249999999999,
        Timestamp('1964-09-01 00:00:00')],
       ['Oct-1964', 9.109, 10.107999999999999, 10.3385,
        Timestamp('1964-10-01 00:00:00')],
       ['Nov-1964', 7.685, 9.042333333333334, 9.50225,
```

```
 Timestamp('1964-11-01 00:00:00')],
['Dec-1964', 7.682, 8.158666666666667, 8.70225,
 Timestamp('1964-12-01 00:00:00')],
['Jan-1965', 8.35, 7.905666666666668, 8.2065,
 Timestamp('1965-01-01 00:00:00')],
['Feb-1965', 7.829, 7.953666666666668, 7.8865,
 Timestamp('1965-02-01 00:00:00')],
['Mar-1965', 8.829, 8.336, 8.1725,
 Timestamp('1965-03-01 00:00:00')],
['Apr-1965', 9.948, 8.868666666666668, 8.739,
 Timestamp('1965-04-01 00:00:00')],
['May-1965', 10.638, 9.805000000000001, 9.311,
 Timestamp('1965-05-01 00:00:00')],
['Jun-1965', 11.253, 10.613000000000001, 10.167,
 Timestamp('1965-06-01 00:00:00')],
['Jul-1965', 11.424, 11.105000000000002, 10.81575,
 Timestamp('1965-07-01 00:00:00')],
['Aug-1965', 11.391, 11.356000000000002, 11.1765,
 Timestamp('1965-08-01 00:00:00')],
['Sep-1965', 10.665, 11.160000000000002, 11.183250000000001,
 Timestamp('1965-09-01 00:00:00')],
['Oct-1965', 9.396, 10.484000000000002, 10.719000000000001,
 Timestamp('1965-10-01 00:00:00')],
['Nov-1965', 7.775, 9.278666666666668, 9.806750000000001,
 Timestamp('1965-11-01 00:00:00')],
['Dec-1965', 7.933, 8.368000000000002, 8.94225,
 Timestamp('1965-12-01 00:00:00')],
['Jan-1966', 8.186, 7.964666666666669, 8.3225,
 Timestamp('1966-01-01 00:00:00')],
['Feb-1966', 7.444, 7.854333333333336, 7.8345,
 Timestamp('1966-02-01 00:00:00')],
['Mar-1966', 8.484, 8.038000000000002, 8.01175,
 Timestamp('1966-03-01 00:00:00')],
['Apr-1966', 9.864, 8.597333333333333, 8.4945,
 Timestamp('1966-04-01 00:00:00')],
['May-1966', 10.252, 9.533333333333335, 9.011000000000001,
 Timestamp('1966-05-01 00:00:00')],
['Jun-1966', 12.282, 10.799333333333335, 10.2205,
 Timestamp('1966-06-01 00:00:00')],
['Jul-1966', 11.637, 11.390333333333336, 11.00875,
 Timestamp('1966-07-01 00:00:00')],
['Aug-1966', 11.577, 11.832, 11.437000000000001,
 Timestamp('1966-08-01 00:00:00')],
['Sep-1966', 12.417, 11.877, 11.97825,
 Timestamp('1966-09-01 00:00:00')],
['Oct-1966', 9.637, 11.210333333333333, 11.317,
 Timestamp('1966-10-01 00:00:00')],
['Nov-1966', 8.094, 10.049333333333335, 10.43125,
 Timestamp('1966-11-01 00:00:00')],
['Dec-1966', 9.28, 9.003666666666666, 9.857,
 Timestamp('1966-12-01 00:00:00')],
['Jan-1967', 8.334, 8.569333333333335, 8.83625,
 Timestamp('1967-01-01 00:00:00')],
```

```
['Feb-1967', 7.899, 8.504333333333333, 8.40175,
 Timestamp('1967-02-01 00:00:00')],
['Mar-1967', 9.994, 8.742333333333335, 8.87675,
 Timestamp('1967-03-01 00:00:00')],
['Apr-1967', 10.078, 9.323666666666666, 9.076249999999998,
 Timestamp('1967-04-01 00:00:00')],
['May-1967', 10.801, 10.291000000000002, 9.693,
 Timestamp('1967-05-01 00:00:00')],
['Jun-1967', 12.953, 11.277333333333333, 10.9565,
 Timestamp('1967-06-01 00:00:00')],
['Jul-1967', 12.222, 11.991999999999999, 11.5135,
 Timestamp('1967-07-01 00:00:00')],
['Aug-1967', 12.246, 12.473666666666666, 12.055499999999999,
 Timestamp('1967-08-01 00:00:00')],
['Sep-1967', 13.281, 12.583000000000004, 12.6755,
 Timestamp('1967-09-01 00:00:00')],
['Oct-1967', 10.366, 11.964333333333334, 12.028749999999999,
 Timestamp('1967-10-01 00:00:00')],
['Nov-1967', 8.73, 10.792333333333334, 11.155750000000001,
 Timestamp('1967-11-01 00:00:00')],
['Dec-1967', 9.614, 9.570000000000002, 10.49775,
 Timestamp('1967-12-01 00:00:00')],
['Jan-1968', 8.639, 8.994333333333335, 9.337250000000001,
 Timestamp('1968-01-01 00:00:00')],
['Feb-1968', 8.772, 9.008333333333335, 8.93875,
 Timestamp('1968-02-01 00:00:00')],
['Mar-1968', 10.894, 9.435, 9.47975,
 Timestamp('1968-03-01 00:00:00')],
['Apr-1968', 10.455, 10.040333333333335, 9.69,
 Timestamp('1968-04-01 00:00:00')],
['May-1968', 11.179, 10.842666666666668, 10.325,
 Timestamp('1968-05-01 00:00:00')],
['Jun-1968', 10.588, 10.740666666666668, 10.779,
 Timestamp('1968-06-01 00:00:00')],
['Jul-1968', 10.794, 10.853666666666669, 10.754,
 Timestamp('1968-07-01 00:00:00')],
['Aug-1968', 12.77, 11.384, 11.332749999999999,
 Timestamp('1968-08-01 00:00:00')],
['Sep-1968', 13.812, 12.458666666666668, 11.991,
 Timestamp('1968-09-01 00:00:00')],
['Oct-1968', 10.857, 12.479666666666668, 12.058250000000001,
 Timestamp('1968-10-01 00:00:00')],
['Nov-1968', 9.29, 11.319666666666668, 11.68225,
 Timestamp('1968-11-01 00:00:00')],
['Dec-1968', 10.925, 10.357333333333335, 11.221,
 Timestamp('1968-12-01 00:00:00')],
['Jan-1969', 9.491, 9.902000000000001, 10.14075,
 Timestamp('1969-01-01 00:00:00')],
['Feb-1969', 8.919, 9.778333333333334, 9.65625,
 Timestamp('1969-02-01 00:00:00')],
['Mar-1969', 11.607, 10.005666666666668, 10.2355,
 Timestamp('1969-03-01 00:00:00')],
['Apr-1969', 8.852, 9.792666666666667, 9.71725,
```

```
 Timestamp('1969-04-01 00:00:00')],
['May-1969', 12.537, 10.998666666666667, 10.47875,
 Timestamp('1969-05-01 00:00:00')],
['Jun-1969', 14.759, 12.049333333333335, 11.938749999999999,
 Timestamp('1969-06-01 00:00:00')],
['Jul-1969', 13.667, 13.654333333333335, 12.45375,
 Timestamp('1969-07-01 00:00:00')],
['Aug-1969', 13.731, 14.052333333333335, 13.6735,
 Timestamp('1969-08-01 00:00:00')],
['Sep-1969', 15.11, 14.169333333333334, 14.31675,
 Timestamp('1969-09-01 00:00:00')],
['Oct-1969', 12.185, 13.675333333333334, 13.67325,
 Timestamp('1969-10-01 00:00:00')],
['Nov-1969', 10.645, 12.646666666666667, 12.91775,
 Timestamp('1969-11-01 00:00:00')],
['Dec-1969', 12.161, 11.663666666666666, 12.52525,
 Timestamp('1969-12-01 00:00:00')]], dtype=object)
```

In [232…

```
def calculate_p_q():
    order_aic_bic = []
    for p in range(4):
        for q in range(4):
            try:
                model = sm.tsa.statespace.SARIMAX(training_data['Miles, in M
                results = model.fit()
                order_aic_bic.append((p, q, results.aic, results.bic))
            except:
                continue
    return order_aic_bic
order_aic_bic = calculate_p_q()
order_df = pd.DataFrame(order_aic_bic, columns=['p', 'q', 'AIC', 'BIC'])
print(order_df.sort_values('AIC'))
print(order_df.sort_values('BIC'))
```

```
/Users/dhavalgarg/opt/anaconda3/lib/python3.9/site-packages/statsmodels/tsa/
statespace/sarimax.py:866: UserWarning: Too few observations to estimate sta
rting parameters for seasonal ARMA. All parameters except for variances will
be set to zeros.
  warn('Too few observations to estimate starting parameters%s.'
 This problem is unconstrained.
```

```
RUNNING THE L-BFGS-B CODE

           * * *

Machine precision = 2.220D-16
 N =            3     M =           10

At X0         0 variables are exactly at the bounds

At iterate    0    f=  1.45847D+00    |proj g|=  7.42216D-02

At iterate    5    f=  1.21765D+00    |proj g|=  4.60395D-02

At iterate   10    f=  1.20223D+00    |proj g|=  4.57948D-03

At iterate   15    f=  1.20156D+00    |proj g|=  2.98707D-03

At iterate   20    f=  1.20146D+00    |proj g|=  1.23112D-03

At iterate   25    f=  1.20144D+00    |proj g|=  1.22053D-04
  ys=-7.837E-07  -gs= 9.270E-07 BFGS update SKIPPED
 Bad direction in the line search;
   refresh the lbfgs memory and restart the iteration.
           * * *

Tit   = total number of iterations
Tnf   = total number of function evaluations
Tnint = total number of segments explored during Cauchy searches
Skip  = number of BFGS updates skipped
Nact  = number of active bounds at final generalized Cauchy point
Projg = norm of the final projected gradient
F     = final function value

           * * *

   N    Tit     Tnf  Tnint  Skip  Nact     Projg        F
    3     28     78      2     1     0   1.779D-06   1.201D+00
  F =   1.2014435116070783

CONVERGENCE: NORM_OF_PROJECTED_GRADIENT_<=_PGTOL
/Users/dhavalgarg/opt/anaconda3/lib/python3.9/site-packages/statsmodels/tsa/
statespace/sarimax.py:866: UserWarning: Too few observations to estimate sta
rting parameters for seasonal ARMA. All parameters except for variances will
be set to zeros.
  warn('Too few observations to estimate starting parameters%s.'
 This problem is unconstrained.
```

```
RUNNING THE L-BFGS-B CODE

           * * *

Machine precision = 2.220D-16
 N =            3     M =           10

At X0         0 variables are exactly at the bounds

At iterate    0    f=  1.45847D+00    |proj g|=  7.42216D-02

At iterate    5    f=  1.21765D+00    |proj g|=  4.60395D-02

At iterate   10    f=  1.20223D+00    |proj g|=  4.57948D-03

At iterate   15    f=  1.20156D+00    |proj g|=  2.98707D-03

At iterate   20    f=  1.20146D+00    |proj g|=  1.23112D-03

At iterate   25    f=  1.20144D+00    |proj g|=  1.22053D-04
  ys=-7.837E-07  -gs= 9.270E-07 BFGS update SKIPPED
 Bad direction in the line search;
   refresh the lbfgs memory and restart the iteration.
/Users/dhavalgarg/opt/anaconda3/lib/python3.9/site-packages/statsmodels/tsa/
statespace/sarimax.py:866: UserWarning: Too few observations to estimate sta
rting parameters for seasonal ARMA. All parameters except for variances will
be set to zeros.
  warn('Too few observations to estimate starting parameters%s.'
 This problem is unconstrained.
```

```
           * * *

Tit   = total number of iterations
Tnf   = total number of function evaluations
Tnint = total number of segments explored during Cauchy searches
Skip  = number of BFGS updates skipped
Nact  = number of active bounds at final generalized Cauchy point
Projg = norm of the final projected gradient
F     = final function value

           * * *

   N    Tit     Tnf  Tnint  Skip  Nact     Projg        F
    3     28     78      2     1     0   1.779D-06   1.201D+00
  F =   1.2014435116070783

CONVERGENCE: NORM_OF_PROJECTED_GRADIENT_<=_PGTOL
RUNNING THE L-BFGS-B CODE

           * * *

Machine precision = 2.220D-16
 N =            3     M =           10

At X0         0 variables are exactly at the bounds

At iterate    0    f=  1.45847D+00    |proj g|=  7.42216D-02

At iterate    5    f=  1.21765D+00    |proj g|=  4.60395D-02

At iterate   10    f=  1.20223D+00    |proj g|=  4.57948D-03

At iterate   15    f=  1.20156D+00    |proj g|=  2.98707D-03

At iterate   20    f=  1.20146D+00    |proj g|=  1.23112D-03

At iterate   25    f=  1.20144D+00    |proj g|=  1.22053D-04
  ys=-7.837E-07  -gs= 9.270E-07 BFGS update SKIPPED
 Bad direction in the line search;
   refresh the lbfgs memory and restart the iteration.
```

```
           * * *

Tit   = total number of iterations
Tnf   = total number of function evaluations
Tnint = total number of segments explored during Cauchy searches
Skip  = number of BFGS updates skipped
Nact  = number of active bounds at final generalized Cauchy point
Projg = norm of the final projected gradient
F     = final function value

           * * *

   N    Tit     Tnf  Tnint  Skip  Nact     Projg        F
    3     28     78      2     1     0   1.779D-06   1.201D+00
  F =   1.2014435116070783

CONVERGENCE: NORM_OF_PROJECTED_GRADIENT_<=_PGTOL
RUNNING THE L-BFGS-B CODE

           * * *

Machine precision = 2.220D-16
 N =            3     M =           10

At X0         0 variables are exactly at the bounds

At iterate    0    f=  1.45847D+00    |proj g|=  7.42216D-02
/Users/dhavalgarg/opt/anaconda3/lib/python3.9/site-packages/statsmodels/tsa/
statespace/sarimax.py:866: UserWarning: Too few observations to estimate sta
rting parameters for seasonal ARMA. All parameters except for variances will
be set to zeros.
  warn('Too few observations to estimate starting parameters%s.'
 This problem is unconstrained.
At iterate    5    f=  1.21765D+00    |proj g|=  4.60395D-02

At iterate   10    f=  1.20223D+00    |proj g|=  4.57948D-03

At iterate   15    f=  1.20156D+00    |proj g|=  2.98707D-03

At iterate   20    f=  1.20146D+00    |proj g|=  1.23112D-03

At iterate   25    f=  1.20144D+00    |proj g|=  1.22053D-04
  ys=-7.837E-07  -gs= 9.270E-07 BFGS update SKIPPED
 Bad direction in the line search;
   refresh the lbfgs memory and restart the iteration.
/Users/dhavalgarg/opt/anaconda3/lib/python3.9/site-packages/statsmodels/tsa/
statespace/sarimax.py:866: UserWarning: Too few observations to estimate sta
rting parameters for seasonal ARMA. All parameters except for variances will
be set to zeros.
  warn('Too few observations to estimate starting parameters%s.'
 This problem is unconstrained.
```

```
           * * *

Tit   = total number of iterations
Tnf   = total number of function evaluations
Tnint = total number of segments explored during Cauchy searches
Skip  = number of BFGS updates skipped
Nact  = number of active bounds at final generalized Cauchy point
Projg = norm of the final projected gradient
F     = final function value

           * * *

   N    Tit     Tnf  Tnint  Skip  Nact     Projg        F
    3     28     78      2     1     0   1.779D-06   1.201D+00
  F =   1.2014435116070783

CONVERGENCE: NORM_OF_PROJECTED_GRADIENT_<=_PGTOL
RUNNING THE L-BFGS-B CODE

           * * *

Machine precision = 2.220D-16
 N =            3     M =           10

At X0         0 variables are exactly at the bounds

At iterate    0    f=  1.45847D+00    |proj g|=  7.42216D-02

At iterate    5    f=  1.21765D+00    |proj g|=  4.60395D-02

At iterate   10    f=  1.20223D+00    |proj g|=  4.57948D-03

At iterate   15    f=  1.20156D+00    |proj g|=  2.98707D-03

At iterate   20    f=  1.20146D+00    |proj g|=  1.23112D-03

At iterate   25    f=  1.20144D+00    |proj g|=  1.22053D-04
  ys=-7.837E-07  -gs= 9.270E-07 BFGS update SKIPPED
 Bad direction in the line search;
   refresh the lbfgs memory and restart the iteration.
/Users/dhavalgarg/opt/anaconda3/lib/python3.9/site-packages/statsmodels/tsa/
statespace/sarimax.py:866: UserWarning: Too few observations to estimate sta
rting parameters for seasonal ARMA. All parameters except for variances will
be set to zeros.
  warn('Too few observations to estimate starting parameters%s.'
 This problem is unconstrained.
```

```
           * * *

Tit   = total number of iterations
Tnf   = total number of function evaluations
Tnint = total number of segments explored during Cauchy searches
Skip  = number of BFGS updates skipped
Nact  = number of active bounds at final generalized Cauchy point
Projg = norm of the final projected gradient
F     = final function value

           * * *

   N    Tit     Tnf  Tnint  Skip  Nact     Projg        F
    3     28     78      2     1     0   1.779D-06   1.201D+00
  F =   1.2014435116070783

CONVERGENCE: NORM_OF_PROJECTED_GRADIENT_<=_PGTOL
RUNNING THE L-BFGS-B CODE

           * * *

Machine precision = 2.220D-16
 N =            3     M =           10

At X0         0 variables are exactly at the bounds

At iterate    0    f=  1.45847D+00    |proj g|=  7.42216D-02

At iterate    5    f=  1.21765D+00    |proj g|=  4.60395D-02

At iterate   10    f=  1.20223D+00    |proj g|=  4.57948D-03

At iterate   15    f=  1.20156D+00    |proj g|=  2.98707D-03

At iterate   20    f=  1.20146D+00    |proj g|=  1.23112D-03

At iterate   25    f=  1.20144D+00    |proj g|=  1.22053D-04
  ys=-7.837E-07  -gs= 9.270E-07 BFGS update SKIPPED
 Bad direction in the line search;
   refresh the lbfgs memory and restart the iteration.
/Users/dhavalgarg/opt/anaconda3/lib/python3.9/site-packages/statsmodels/tsa/
statespace/sarimax.py:866: UserWarning: Too few observations to estimate sta
rting parameters for seasonal ARMA. All parameters except for variances will
be set to zeros.
  warn('Too few observations to estimate starting parameters%s.'
 This problem is unconstrained.
```

```
           * * *

Tit   = total number of iterations
Tnf   = total number of function evaluations
Tnint = total number of segments explored during Cauchy searches
Skip  = number of BFGS updates skipped
Nact  = number of active bounds at final generalized Cauchy point
Projg = norm of the final projected gradient
F     = final function value

           * * *

   N    Tit     Tnf  Tnint  Skip  Nact     Projg        F
    3     28     78      2     1     0   1.779D-06   1.201D+00
  F =   1.2014435116070783

CONVERGENCE: NORM_OF_PROJECTED_GRADIENT_<=_PGTOL
RUNNING THE L-BFGS-B CODE

           * * *

Machine precision = 2.220D-16
 N =            3     M =           10

At X0         0 variables are exactly at the bounds

At iterate    0    f=  1.45847D+00    |proj g|=  7.42216D-02

At iterate    5    f=  1.21765D+00    |proj g|=  4.60395D-02

At iterate   10    f=  1.20223D+00    |proj g|=  4.57948D-03

At iterate   15    f=  1.20156D+00    |proj g|=  2.98707D-03

At iterate   20    f=  1.20146D+00    |proj g|=  1.23112D-03

At iterate   25    f=  1.20144D+00    |proj g|=  1.22053D-04
  ys=-7.837E-07  -gs= 9.270E-07 BFGS update SKIPPED
 Bad direction in the line search;
   refresh the lbfgs memory and restart the iteration.
/Users/dhavalgarg/opt/anaconda3/lib/python3.9/site-packages/statsmodels/tsa/
statespace/sarimax.py:866: UserWarning: Too few observations to estimate sta
rting parameters for seasonal ARMA. All parameters except for variances will
be set to zeros.
  warn('Too few observations to estimate starting parameters%s.'
 This problem is unconstrained.
```

```
           * * *

Tit   = total number of iterations
Tnf   = total number of function evaluations
Tnint = total number of segments explored during Cauchy searches
Skip  = number of BFGS updates skipped
Nact  = number of active bounds at final generalized Cauchy point
Projg = norm of the final projected gradient
F     = final function value

           * * *

   N    Tit     Tnf  Tnint  Skip  Nact     Projg        F
    3     28     78      2     1     0   1.779D-06   1.201D+00
  F =   1.2014435116070783

CONVERGENCE: NORM_OF_PROJECTED_GRADIENT_<=_PGTOL
RUNNING THE L-BFGS-B CODE

           * * *

Machine precision = 2.220D-16
 N =            3     M =           10

At X0         0 variables are exactly at the bounds

At iterate    0    f=  1.45847D+00    |proj g|=  7.42216D-02

At iterate    5    f=  1.21765D+00    |proj g|=  4.60395D-02

At iterate   10    f=  1.20223D+00    |proj g|=  4.57948D-03

At iterate   15    f=  1.20156D+00    |proj g|=  2.98707D-03

At iterate   20    f=  1.20146D+00    |proj g|=  1.23112D-03

At iterate   25    f=  1.20144D+00    |proj g|=  1.22053D-04
  ys=-7.837E-07  -gs= 9.270E-07 BFGS update SKIPPED
 Bad direction in the line search;
   refresh the lbfgs memory and restart the iteration.
/Users/dhavalgarg/opt/anaconda3/lib/python3.9/site-packages/statsmodels/tsa/
statespace/sarimax.py:866: UserWarning: Too few observations to estimate sta
rting parameters for seasonal ARMA. All parameters except for variances will
be set to zeros.
  warn('Too few observations to estimate starting parameters%s.'
 This problem is unconstrained.
```

```
           * * *

Tit   = total number of iterations
Tnf   = total number of function evaluations
Tnint = total number of segments explored during Cauchy searches
Skip  = number of BFGS updates skipped
Nact  = number of active bounds at final generalized Cauchy point
Projg = norm of the final projected gradient
F     = final function value

           * * *

   N    Tit     Tnf  Tnint  Skip  Nact     Projg        F
    3     28     78      2     1     0   1.779D-06   1.201D+00
  F =   1.2014435116070783

CONVERGENCE: NORM_OF_PROJECTED_GRADIENT_<=_PGTOL
RUNNING THE L-BFGS-B CODE

           * * *

Machine precision = 2.220D-16
 N =            3     M =           10

At X0         0 variables are exactly at the bounds

At iterate    0    f=  1.45847D+00    |proj g|=  7.42216D-02

At iterate    5    f=  1.21765D+00    |proj g|=  4.60395D-02

At iterate   10    f=  1.20223D+00    |proj g|=  4.57948D-03

At iterate   15    f=  1.20156D+00    |proj g|=  2.98707D-03

At iterate   20    f=  1.20146D+00    |proj g|=  1.23112D-03

At iterate   25    f=  1.20144D+00    |proj g|=  1.22053D-04
  ys=-7.837E-07  -gs= 9.270E-07 BFGS update SKIPPED
 Bad direction in the line search;
   refresh the lbfgs memory and restart the iteration.
/Users/dhavalgarg/opt/anaconda3/lib/python3.9/site-packages/statsmodels/tsa/
statespace/sarimax.py:866: UserWarning: Too few observations to estimate sta
rting parameters for seasonal ARMA. All parameters except for variances will
be set to zeros.
  warn('Too few observations to estimate starting parameters%s.'
 This problem is unconstrained.
```

```
           * * *

Tit   = total number of iterations
Tnf   = total number of function evaluations
Tnint = total number of segments explored during Cauchy searches
Skip  = number of BFGS updates skipped
Nact  = number of active bounds at final generalized Cauchy point
Projg = norm of the final projected gradient
F     = final function value

           * * *

   N    Tit     Tnf  Tnint  Skip  Nact     Projg        F
    3     28     78      2     1     0   1.779D-06   1.201D+00
  F =   1.2014435116070783

CONVERGENCE: NORM_OF_PROJECTED_GRADIENT_<=_PGTOL
RUNNING THE L-BFGS-B CODE

           * * *

Machine precision = 2.220D-16
 N =            3     M =           10

At X0         0 variables are exactly at the bounds

At iterate    0    f=  1.45847D+00    |proj g|=  7.42216D-02

At iterate    5    f=  1.21765D+00    |proj g|=  4.60395D-02

At iterate   10    f=  1.20223D+00    |proj g|=  4.57948D-03

At iterate   15    f=  1.20156D+00    |proj g|=  2.98707D-03

At iterate   20    f=  1.20146D+00    |proj g|=  1.23112D-03

At iterate   25    f=  1.20144D+00    |proj g|=  1.22053D-04
  ys=-7.837E-07  -gs= 9.270E-07 BFGS update SKIPPED
 Bad direction in the line search;
   refresh the lbfgs memory and restart the iteration.
/Users/dhavalgarg/opt/anaconda3/lib/python3.9/site-packages/statsmodels/tsa/
statespace/sarimax.py:866: UserWarning: Too few observations to estimate sta
rting parameters for seasonal ARMA. All parameters except for variances will
be set to zeros.
  warn('Too few observations to estimate starting parameters%s.'
 This problem is unconstrained.
```

```
           * * *

Tit   = total number of iterations
Tnf   = total number of function evaluations
Tnint = total number of segments explored during Cauchy searches
Skip  = number of BFGS updates skipped
Nact  = number of active bounds at final generalized Cauchy point
Projg = norm of the final projected gradient
F     = final function value

           * * *

   N    Tit     Tnf  Tnint  Skip  Nact     Projg        F
    3     28     78      2     1     0   1.779D-06   1.201D+00
  F =   1.2014435116070783

CONVERGENCE: NORM_OF_PROJECTED_GRADIENT_<=_PGTOL
RUNNING THE L-BFGS-B CODE

           * * *

Machine precision = 2.220D-16
 N =            3     M =           10

At X0         0 variables are exactly at the bounds

At iterate    0    f=  1.45847D+00    |proj g|=  7.42216D-02

At iterate    5    f=  1.21765D+00    |proj g|=  4.60395D-02

At iterate   10    f=  1.20223D+00    |proj g|=  4.57948D-03

At iterate   15    f=  1.20156D+00    |proj g|=  2.98707D-03

At iterate   20    f=  1.20146D+00    |proj g|=  1.23112D-03

At iterate   25    f=  1.20144D+00    |proj g|=  1.22053D-04
  ys=-7.837E-07  -gs= 9.270E-07 BFGS update SKIPPED
 Bad direction in the line search;
   refresh the lbfgs memory and restart the iteration.
/Users/dhavalgarg/opt/anaconda3/lib/python3.9/site-packages/statsmodels/tsa/
statespace/sarimax.py:866: UserWarning: Too few observations to estimate sta
rting parameters for seasonal ARMA. All parameters except for variances will
be set to zeros.
  warn('Too few observations to estimate starting parameters%s.'
 This problem is unconstrained.
```

```
           * * *

Tit   = total number of iterations
Tnf   = total number of function evaluations
Tnint = total number of segments explored during Cauchy searches
Skip  = number of BFGS updates skipped
Nact  = number of active bounds at final generalized Cauchy point
Projg = norm of the final projected gradient
F     = final function value

           * * *

   N    Tit     Tnf  Tnint  Skip  Nact     Projg        F
    3     28     78      2     1     0   1.779D-06   1.201D+00
  F =   1.2014435116070783

CONVERGENCE: NORM_OF_PROJECTED_GRADIENT_<=_PGTOL
RUNNING THE L-BFGS-B CODE

           * * *

Machine precision = 2.220D-16
 N =            3     M =           10

At X0         0 variables are exactly at the bounds

At iterate    0    f=  1.45847D+00    |proj g|=  7.42216D-02

At iterate    5    f=  1.21765D+00    |proj g|=  4.60395D-02

At iterate   10    f=  1.20223D+00    |proj g|=  4.57948D-03

At iterate   15    f=  1.20156D+00    |proj g|=  2.98707D-03

At iterate   20    f=  1.20146D+00    |proj g|=  1.23112D-03

At iterate   25    f=  1.20144D+00    |proj g|=  1.22053D-04
  ys=-7.837E-07  -gs= 9.270E-07 BFGS update SKIPPED
 Bad direction in the line search;
   refresh the lbfgs memory and restart the iteration.
/Users/dhavalgarg/opt/anaconda3/lib/python3.9/site-packages/statsmodels/tsa/
statespace/sarimax.py:866: UserWarning: Too few observations to estimate sta
rting parameters for seasonal ARMA. All parameters except for variances will
be set to zeros.
  warn('Too few observations to estimate starting parameters%s.'
 This problem is unconstrained.
```

```
           * * *

Tit   = total number of iterations
Tnf   = total number of function evaluations
Tnint = total number of segments explored during Cauchy searches
Skip  = number of BFGS updates skipped
Nact  = number of active bounds at final generalized Cauchy point
Projg = norm of the final projected gradient
F     = final function value

           * * *

   N    Tit     Tnf  Tnint  Skip  Nact     Projg        F
    3     28     78      2     1     0   1.779D-06   1.201D+00
  F =   1.2014435116070783

CONVERGENCE: NORM_OF_PROJECTED_GRADIENT_<=_PGTOL
RUNNING THE L-BFGS-B CODE

           * * *

Machine precision = 2.220D-16
 N =            3     M =           10

At X0         0 variables are exactly at the bounds

At iterate    0    f=  1.45847D+00    |proj g|=  7.42216D-02

At iterate    5    f=  1.21765D+00    |proj g|=  4.60395D-02

At iterate   10    f=  1.20223D+00    |proj g|=  4.57948D-03

At iterate   15    f=  1.20156D+00    |proj g|=  2.98707D-03

At iterate   20    f=  1.20146D+00    |proj g|=  1.23112D-03

At iterate   25    f=  1.20144D+00    |proj g|=  1.22053D-04
  ys=-7.837E-07  -gs= 9.270E-07 BFGS update SKIPPED
 Bad direction in the line search;
   refresh the lbfgs memory and restart the iteration.
/Users/dhavalgarg/opt/anaconda3/lib/python3.9/site-packages/statsmodels/tsa/
statespace/sarimax.py:866: UserWarning: Too few observations to estimate sta
rting parameters for seasonal ARMA. All parameters except for variances will
be set to zeros.
  warn('Too few observations to estimate starting parameters%s.'
 This problem is unconstrained.
```

```
           * * *

Tit   = total number of iterations
Tnf   = total number of function evaluations
Tnint = total number of segments explored during Cauchy searches
Skip  = number of BFGS updates skipped
Nact  = number of active bounds at final generalized Cauchy point
Projg = norm of the final projected gradient
F     = final function value

           * * *

   N    Tit     Tnf  Tnint  Skip  Nact     Projg        F
    3     28     78      2     1     0   1.779D-06   1.201D+00
  F =   1.2014435116070783

CONVERGENCE: NORM_OF_PROJECTED_GRADIENT_<=_PGTOL
RUNNING THE L-BFGS-B CODE

           * * *

Machine precision = 2.220D-16
 N =            3     M =           10

At X0         0 variables are exactly at the bounds

At iterate    0    f=  1.45847D+00    |proj g|=  7.42216D-02

At iterate    5    f=  1.21765D+00    |proj g|=  4.60395D-02

At iterate   10    f=  1.20223D+00    |proj g|=  4.57948D-03

At iterate   15    f=  1.20156D+00    |proj g|=  2.98707D-03

At iterate   20    f=  1.20146D+00    |proj g|=  1.23112D-03

At iterate   25    f=  1.20144D+00    |proj g|=  1.22053D-04
  ys=-7.837E-07  -gs= 9.270E-07 BFGS update SKIPPED
 Bad direction in the line search;
   refresh the lbfgs memory and restart the iteration.
/Users/dhavalgarg/opt/anaconda3/lib/python3.9/site-packages/statsmodels/tsa/
statespace/sarimax.py:866: UserWarning: Too few observations to estimate sta
rting parameters for seasonal ARMA. All parameters except for variances will
be set to zeros.
  warn('Too few observations to estimate starting parameters%s.'
 This problem is unconstrained.
```

```
           * * *

Tit   = total number of iterations
Tnf   = total number of function evaluations
Tnint = total number of segments explored during Cauchy searches
Skip  = number of BFGS updates skipped
Nact  = number of active bounds at final generalized Cauchy point
Projg = norm of the final projected gradient
F     = final function value

           * * *

   N    Tit     Tnf  Tnint  Skip  Nact     Projg        F
    3     28     78      2     1     0   1.779D-06   1.201D+00
  F =   1.2014435116070783

CONVERGENCE: NORM_OF_PROJECTED_GRADIENT_<=_PGTOL
RUNNING THE L-BFGS-B CODE

           * * *

Machine precision = 2.220D-16
 N =            3     M =           10

At X0         0 variables are exactly at the bounds

At iterate    0    f=  1.45847D+00    |proj g|=  7.42216D-02

At iterate    5    f=  1.21765D+00    |proj g|=  4.60395D-02

At iterate   10    f=  1.20223D+00    |proj g|=  4.57948D-03

At iterate   15    f=  1.20156D+00    |proj g|=  2.98707D-03

At iterate   20    f=  1.20146D+00    |proj g|=  1.23112D-03

At iterate   25    f=  1.20144D+00    |proj g|=  1.22053D-04
  ys=-7.837E-07  -gs= 9.270E-07 BFGS update SKIPPED
 Bad direction in the line search;
   refresh the lbfgs memory and restart the iteration.
/Users/dhavalgarg/opt/anaconda3/lib/python3.9/site-packages/statsmodels/tsa/
statespace/sarimax.py:866: UserWarning: Too few observations to estimate sta
rting parameters for seasonal ARMA. All parameters except for variances will
be set to zeros.
  warn('Too few observations to estimate starting parameters%s.'
 This problem is unconstrained.
           * * *

Tit   = total number of iterations
Tnf   = total number of function evaluations
Tnint = total number of segments explored during Cauchy searches
Skip  = number of BFGS updates skipped
Nact  = number of active bounds at final generalized Cauchy point
```

```
Projg = norm of the final projected gradient
F     = final function value

           * * *

   N    Tit     Tnf  Tnint  Skip  Nact     Projg        F
    3     28     78      2     1     0   1.779D-06   1.201D+00
  F =   1.2014435116070783

CONVERGENCE: NORM_OF_PROJECTED_GRADIENT_<=_PGTOL
RUNNING THE L-BFGS-B CODE

           * * *

Machine precision = 2.220D-16
 N =            3     M =           10

At X0         0 variables are exactly at the bounds

At iterate    0    f=  1.45847D+00    |proj g|=  7.42216D-02

At iterate    5    f=  1.21765D+00    |proj g|=  4.60395D-02

At iterate   10    f=  1.20223D+00    |proj g|=  4.57948D-03

At iterate   15    f=  1.20156D+00    |proj g|=  2.98707D-03

At iterate   20    f=  1.20146D+00    |proj g|=  1.23112D-03

At iterate   25    f=  1.20144D+00    |proj g|=  1.22053D-04
  ys=-7.837E-07  -gs= 9.270E-07 BFGS update SKIPPED

           * * *

Tit   = total number of iterations
Tnf   = total number of function evaluations
Tnint = total number of segments explored during Cauchy searches
Skip  = number of BFGS updates skipped
Nact  = number of active bounds at final generalized Cauchy point
Projg = norm of the final projected gradient
F     = final function value

           * * *

   N    Tit     Tnf  Tnint  Skip  Nact     Projg        F
    3     28     78      2     1     0   1.779D-06   1.201D+00
  F =   1.2014435116070783

CONVERGENCE: NORM_OF_PROJECTED_GRADIENT_<=_PGTOL
    p  q         AIC         BIC
0   0  0  179.007866  183.497388
1   0  1  179.007866  183.497388
2   0  2  179.007866  183.497388
```

```
3    0  3  179.007866  183.497388
4    1  0  179.007866  183.497388
5    1  1  179.007866  183.497388
6    1  2  179.007866  183.497388
7    1  3  179.007866  183.497388
8    2  0  179.007866  183.497388
9    2  1  179.007866  183.497388
10   2  2  179.007866  183.497388
11   2  3  179.007866  183.497388
12   3  0  179.007866  183.497388
13   3  1  179.007866  183.497388
14   3  2  179.007866  183.497388
15   3  3  179.007866  183.497388
     p  q         AIC         BIC
0    0  0  179.007866  183.497388
1    0  1  179.007866  183.497388
2    0  2  179.007866  183.497388
3    0  3  179.007866  183.497388
4    1  0  179.007866  183.497388
5    1  1  179.007866  183.497388
6    1  2  179.007866  183.497388
7    1  3  179.007866  183.497388
8    2  0  179.007866  183.497388
9    2  1  179.007866  183.497388
10   2  2  179.007866  183.497388
11   2  3  179.007866  183.497388
12   3  0  179.007866  183.497388
13   3  1  179.007866  183.497388
14   3  2  179.007866  183.497388
15   3  3  179.007866  183.497388
 Bad direction in the line search;
   refresh the lbfgs memory and restart the iteration.
```

We observe that model of order=(3,1,2), seasonal_order=(3,1,2,12) seems to be a good fit as the data is approx. normally distributed

In [241…

```
model=sm.tsa.statespace.SARIMAX(training_data["Miles, in Millions"], order =
results=model.fit()
df['forecast']=results.predict(start=73,end=84,dynamic=True)
df[['Miles, in Millions','forecast']].plot(figsize=(12,8))
results.summary()
```

```
/Users/dhavalgarg/opt/anaconda3/lib/python3.9/site-packages/statsmodels/tsa/
statespace/sarimax.py:978: UserWarning: Non-invertible starting MA parameter
s found. Using zeros as starting parameters.
  warn('Non-invertible starting MA parameters found.'
/Users/dhavalgarg/opt/anaconda3/lib/python3.9/site-packages/statsmodels/tsa/
statespace/sarimax.py:866: UserWarning: Too few observations to estimate sta
rting parameters for seasonal ARMA. All parameters except for variances will
be set to zeros.
  warn('Too few observations to estimate starting parameters%s.'
 This problem is unconstrained.
```

```
RUNNING THE L-BFGS-B CODE

           * * *

Machine precision = 2.220D-16
 N =           11     M =           10

At X0         0 variables are exactly at the bounds

At iterate    0    f=  1.19660D+00    |proj g|=  4.80597D-01

At iterate    5    f=  9.75391D-01    |proj g|=  3.48860D-02

At iterate   10    f=  9.66705D-01    |proj g|=  8.61401D-03

At iterate   15    f=  9.63328D-01    |proj g|=  6.80125D-02

At iterate   20    f=  9.61443D-01    |proj g|=  5.39761D-03

At iterate   25    f=  9.60650D-01    |proj g|=  5.18187D-03

At iterate   30    f=  9.60470D-01    |proj g|=  2.36152D-03

At iterate   35    f=  9.60410D-01    |proj g|=  6.15600D-04

At iterate   40    f=  9.60385D-01    |proj g|=  1.20860D-03

At iterate   45    f=  9.60367D-01    |proj g|=  2.61715D-03
/Users/dhavalgarg/opt/anaconda3/lib/python3.9/site-packages/statsmodels/base
/model.py:604: ConvergenceWarning: Maximum Likelihood optimization failed to
converge. Check mle_retvals
  warnings.warn("Maximum Likelihood optimization failed to "
At iterate   50    f=  9.60342D-01    |proj g|=  7.98183D-04

           * * *

Tit   = total number of iterations
Tnf   = total number of function evaluations
Tnint = total number of segments explored during Cauchy searches
Skip  = number of BFGS updates skipped
Nact  = number of active bounds at final generalized Cauchy point
Projg = norm of the final projected gradient
F     = final function value

           * * *

   N    Tit     Tnf  Tnint  Skip  Nact     Projg        F
   11     50     56      1     0     0   7.982D-04   9.603D-01
  F =  0.96034181236898852

STOP: TOTAL NO. of ITERATIONS REACHED LIMIT
```

```
/Users/dhavalgarg/opt/anaconda3/lib/python3.9/site-packages/statsmodels/tsa/statespace/kalman_filter.py:2290: ValueWarning: Dynamic prediction specified to begin during out-of-sample forecasting period, and so has no effect.
  warn('Dynamic prediction specified to begin during'
```

Out[241]:

SARIMAX Results

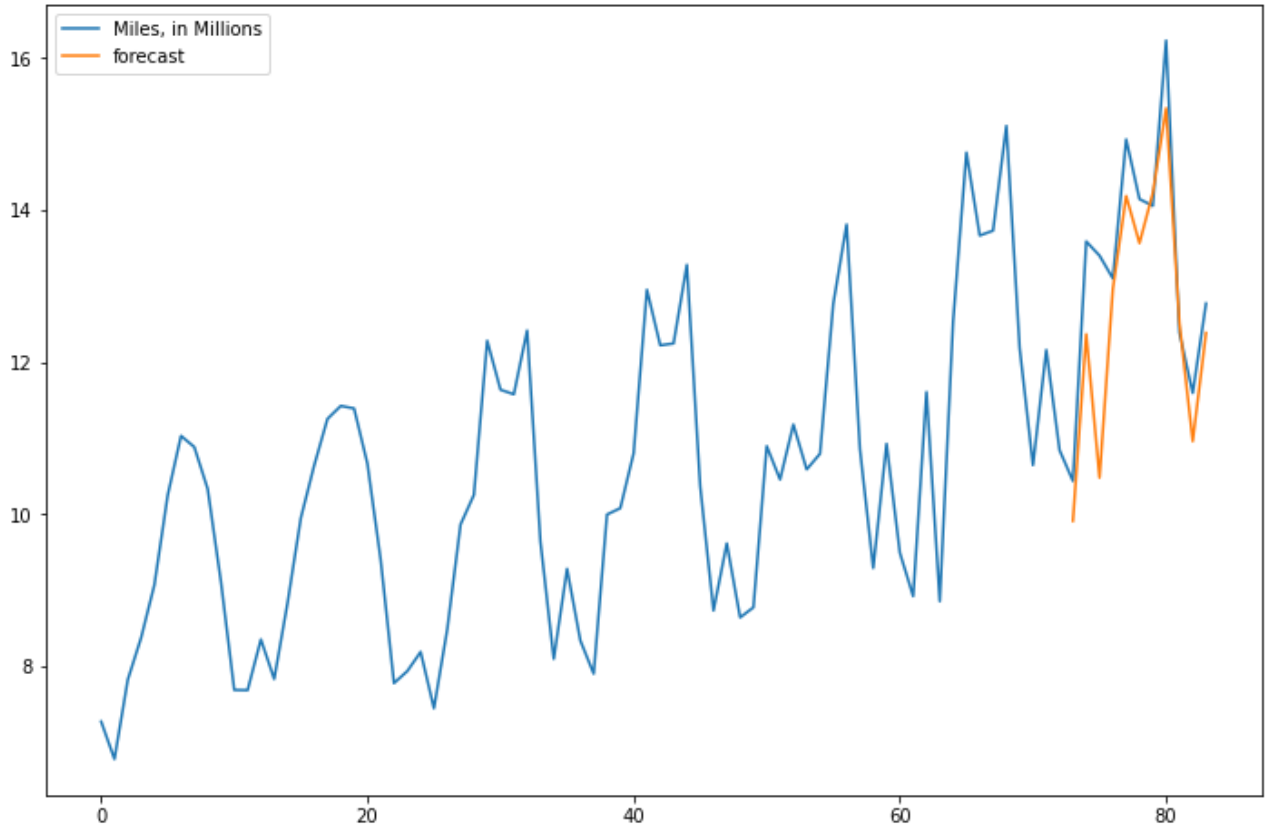
Dep. Variable:	Miles, in Millions	**No. Observations:**	72
Model:	SARIMAX(3, 1, 2)x(3, 1, 2, 12)	**Log Likelihood**	-69.145
Date:	Mon, 31 Oct 2022	**AIC**	160.289
Time:	23:48:02	**BIC**	183.142
Sample:	0	**HQIC**	169.210
	- 72		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.0946	0.769	-0.123	0.902	-1.602	1.413
ar.L2	0.0539	0.484	0.111	0.911	-0.894	1.002
ar.L3	0.2049	0.358	0.572	0.567	-0.497	0.907
ma.L1	-0.4597	17.494	-0.026	0.979	-34.748	33.828
ma.L2	-0.5390	9.335	-0.058	0.954	-18.836	17.758
ar.S.L12	-0.3944	293.474	-0.001	0.999	-575.592	574.803
ar.S.L24	0.8346	138.076	0.006	0.995	-269.789	271.458
ar.S.L36	0.3465	32.092	0.011	0.991	-62.552	63.245
ma.S.L12	-0.0069	1348.669	-5.1e-06	1.000	-2643.350	2643.336
ma.S.L24	-0.9794	413.236	-0.002	0.998	-810.907	808.949
sigma2	0.5190	204.368	0.003	0.998	-400.035	401.074

Ljung-Box (L1) (Q):	0.00	**Jarque-Bera (JB):**	44.67
Prob(Q):	0.98	**Prob(JB):**	0.00
Heteroskedasticity (H):	5.16	**Skew:**	-0.81
Prob(H) (two-sided):	0.00	**Kurtosis:**	6.94

Warnings:

[1] Covariance matrix calculated using the outer product of gradients (complex-step).



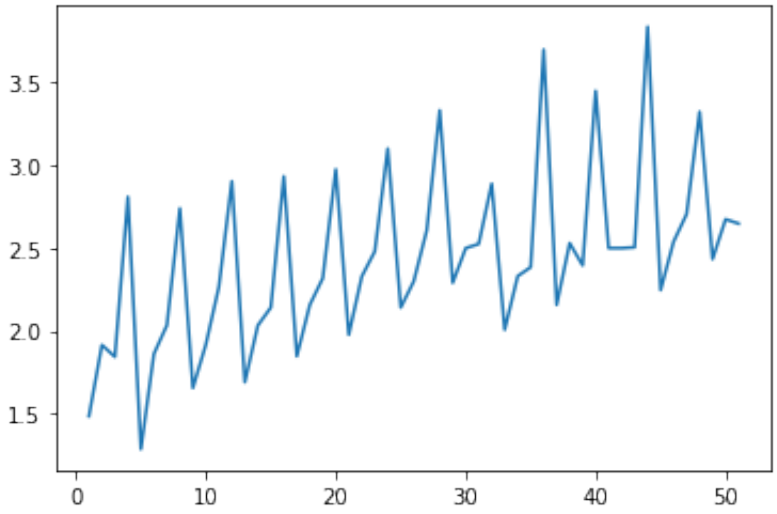
In [180…
```
wine_data = pd.read_csv('TotalWine.csv')
wine_data.head()
```

Out[180]:

	Time (Quarter)	TotalWine
0	1	1.486
1	2	1.915
2	3	1.844
3	4	2.808
4	5	1.287

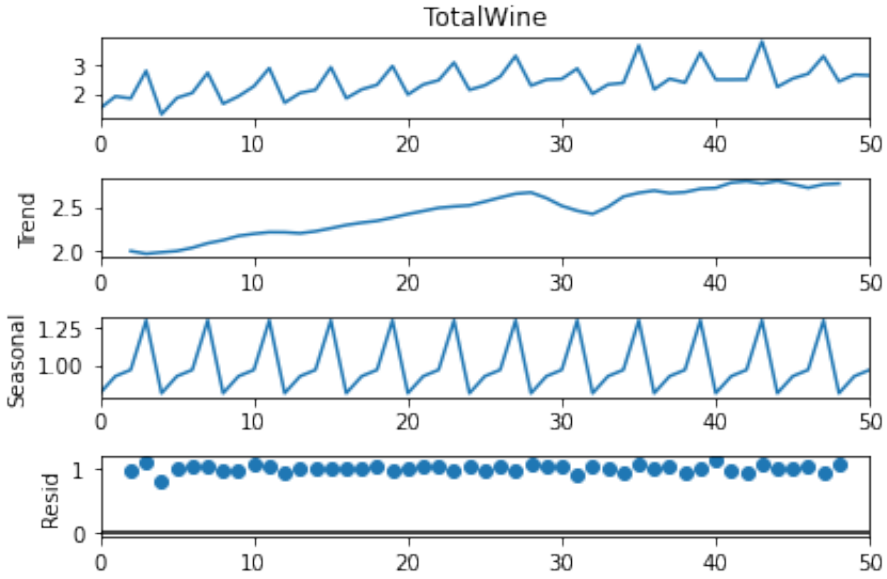
In [181…
```
x = wine_data['Time (Quarter)']
y = wine_data['TotalWine']
plt.plot(x,y)
```

Out[181]:
```
[<matplotlib.lines.Line2D at 0x7fb9903a0400>]
```



In [182…
```
from statsmodels.tsa.seasonal import seasonal_decompose
seasonal = seasonal_decompose(wine_data['TotalWine'],period=4,model='multipl
```

In [183…
```
seasonal.plot();
```



As it can be observed from the graph above the seasonal period for this Time Series is 4.

In [184…
```
wd2 = wine_data[['Time (Quarter)', 'TotalWine']].set_index(['Time (Quarter)'
```

In [185…
```
wd2
```

Out[185]:

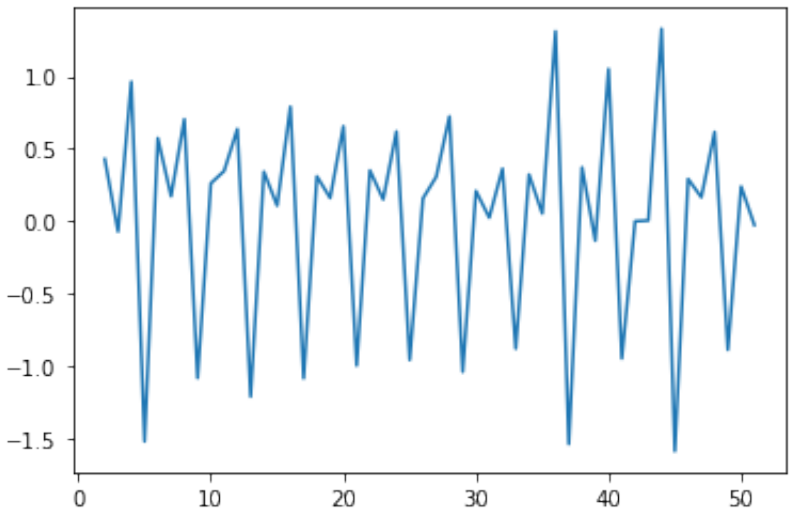
Time (Quarter)	TotalWine
1	1.486

2	1.915
3	1.844
4	2.808
5	1.287
6	1.861
7	2.034
8	2.739
9	1.656
10	1.918
11	2.265
12	2.902
13	1.691
14	2.033
15	2.141
16	2.932
17	1.847
18	2.157
19	2.318
20	2.974
21	1.977
22	2.328
23	2.479
24	3.099
25	2.141
26	2.299
27	2.606
28	3.330
29	2.290
30	2.499
31	2.524
32	2.887
33	2.007

34	2.330
35	2.384
36	3.696
37	2.157
38	2.529
39	2.395
40	3.447
41	2.499
42	2.499
43	2.504
44	3.834
45	2.246
46	2.538
47	2.704
48	3.321
49	2.433
50	2.673
51	2.647

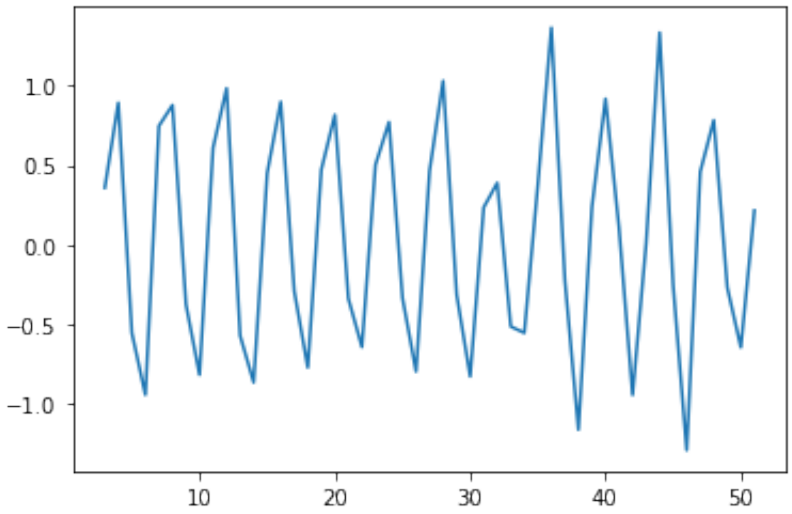
In [186…
```
diff1_wd = wd2.diff()
plt.plot(diff1_wd)
```

Out[186]: [<matplotlib.lines.Line2D at 0x7fb9b84841f0>]



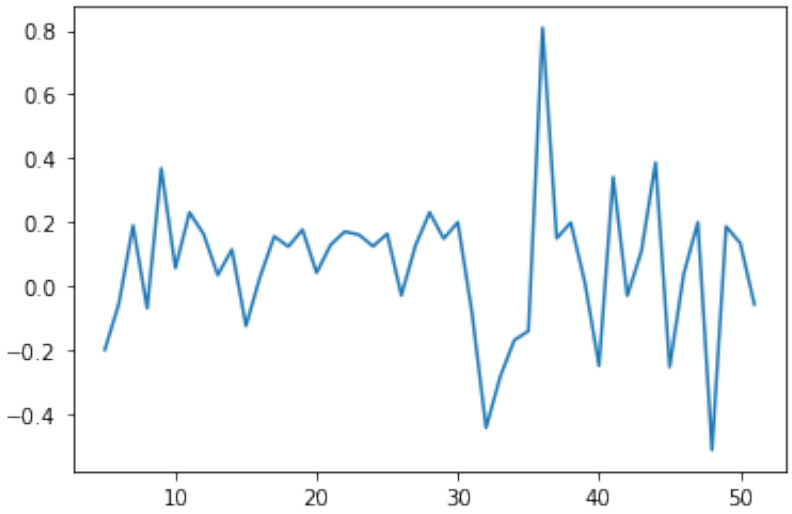
In [187…
```
diff2_wd = wd2.diff(2)
plt.plot(diff2_wd)
```

Out[187]:
```
[<matplotlib.lines.Line2D at 0x7fb9b85f1730>]
```



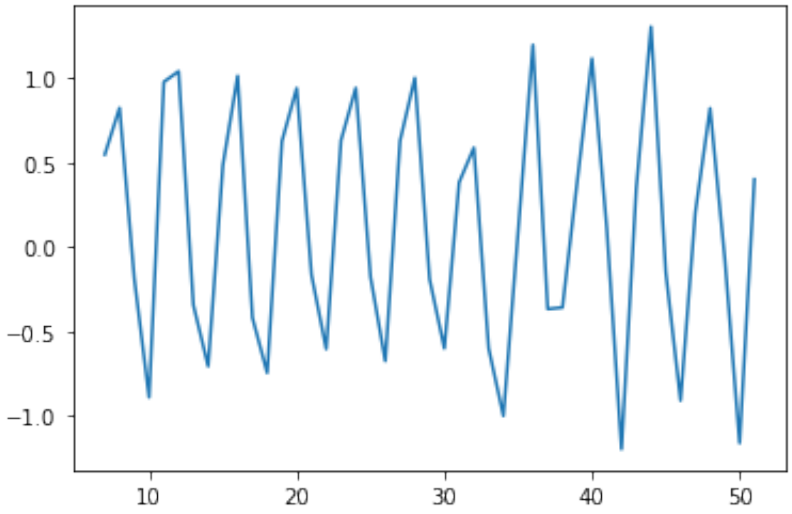
In [188…
```
diff4_wd = wd2.diff(4)
plt.plot(diff4_wd)
```

Out[188]:
```
[<matplotlib.lines.Line2D at 0x7fb9b86db130>]
```



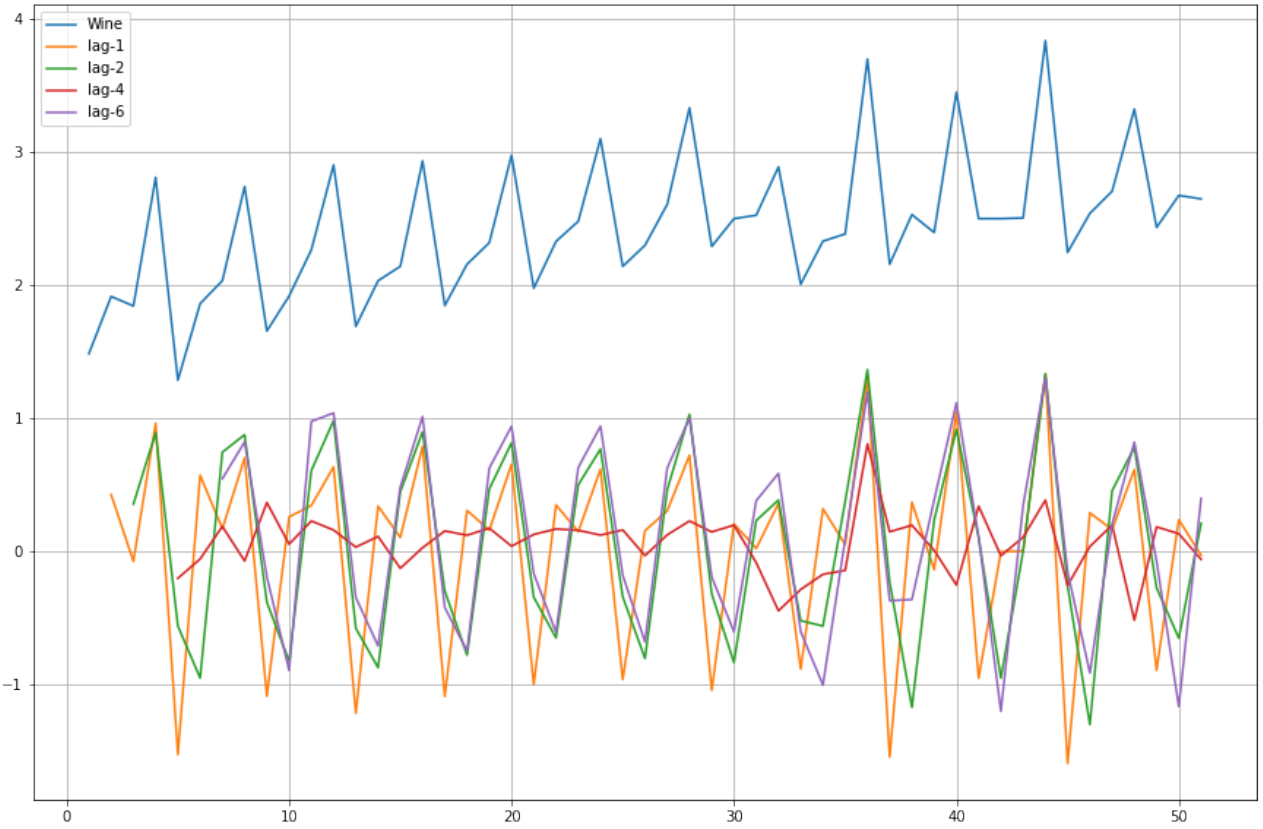
In [189…
```
diff6_wd = wd2.diff(6)
plt.plot(diff6_wd)
```

Out[189]:
```
[<matplotlib.lines.Line2D at 0x7fb9b8735970>]
```

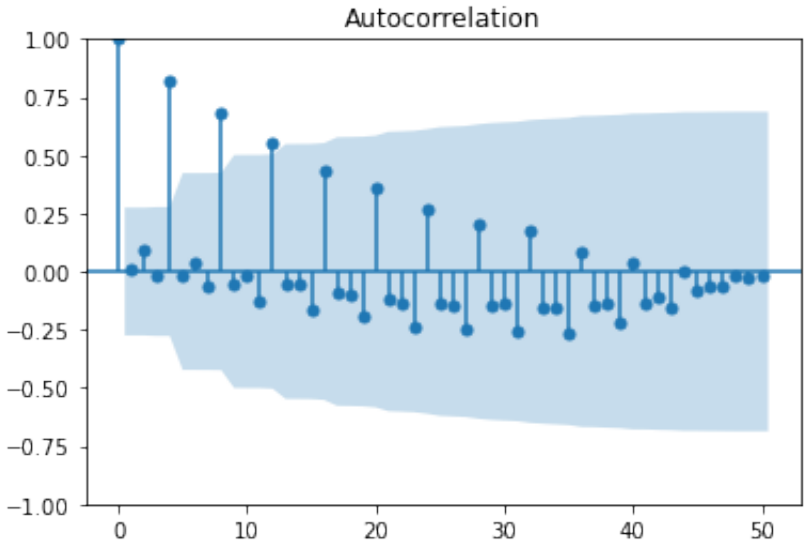


In [192…
```
plt.figure(figsize=(15,10))
plt.grid(True)
plt.plot(wd2['TotalWine'],label='Wine')
plt.plot(diff1_wd['TotalWine'],label='lag-1')
plt.plot(diff2_wd['TotalWine'],label='lag-2')
plt.plot(diff4_wd['TotalWine'],label='lag-4')
plt.plot(diff6_wd['TotalWine'],label='lag-6')
plt.legend(loc=2)
```

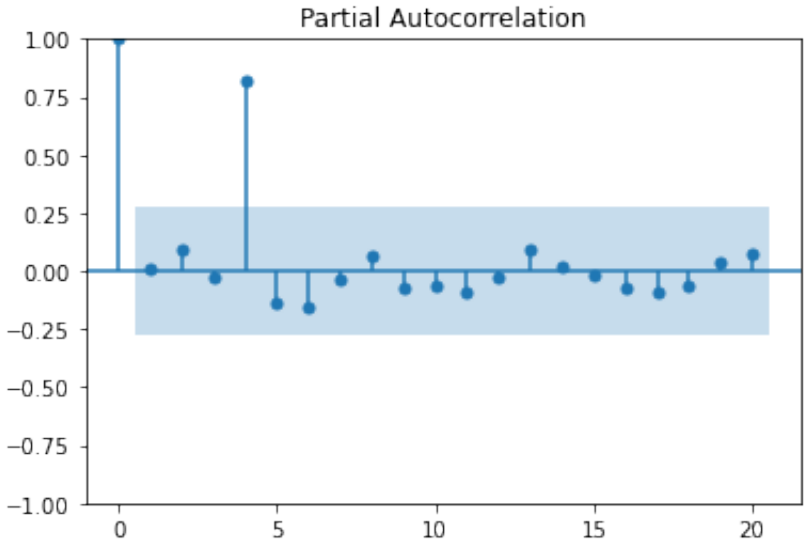
Out[192]: <matplotlib.legend.Legend at 0x7fb9b7044580>



In [193…
```
sm.graphics.tsa.plot_acf(wine_data['TotalWine'],lags=50);
```



In [195…
```
sm.graphics.tsa.plot_pacf(wine_data['TotalWine'],lags=20,method='ywm');
```



In [101…
```
import statsmodels.api as sm
from statsmodels.tsa.ar_model import AutoReg, ar_select_order
```

In [201…
```
diff4_wd.dropna(inplace=True)
```

In [215…
```
diff4_wd.head()
```

Out[215]:

Time (Quarter)	TotalWine
5	-0.199
6	-0.054
7	0.190
8	-0.069
9	0.369

In [214…
```
from sklearn.model_selection import train_test_split
```

In [216…
```
y = diff4_wd['TotalWine']
X = diff4_wd['Time (Quarter)']
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
File ~/opt/anaconda3/lib/python3.9/site-packages/pandas/core/indexes/base.py
:3621, in Index.get_loc(self, key, method, tolerance)
   3620 try:
-> 3621     return self._engine.get_loc(casted_key)
   3622 except KeyError as err:

File ~/opt/anaconda3/lib/python3.9/site-packages/pandas/_libs/index.pyx:136,
in pandas._libs.index.IndexEngine.get_loc()

File ~/opt/anaconda3/lib/python3.9/site-packages/pandas/_libs/index.pyx:163,
in pandas._libs.index.IndexEngine.get_loc()

File pandas/_libs/hashtable_class_helper.pxi:5198, in pandas._libs.hashtable
.PyObjectHashTable.get_item()

File pandas/_libs/hashtable_class_helper.pxi:5206, in pandas._libs.hashtable
.PyObjectHashTable.get_item()

KeyError: 'Time (Quarter)'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
Input In [216], in <cell line: 2>()
      1 y = diff4_wd['TotalWine']
----> 2 X = diff4_wd['Time (Quarter)']

File ~/opt/anaconda3/lib/python3.9/site-packages/pandas/core/frame.py:3505,
in DataFrame.__getitem__(self, key)
   3503 if self.columns.nlevels > 1:
   3504     return self._getitem_multilevel(key)
-> 3505 indexer = self.columns.get_loc(key)
   3506 if is_integer(indexer):
   3507     indexer = [indexer]

File ~/opt/anaconda3/lib/python3.9/site-packages/pandas/core/indexes/base.py
:3623, in Index.get_loc(self, key, method, tolerance)
   3621     return self._engine.get_loc(casted_key)
   3622 except KeyError as err:
-> 3623     raise KeyError(key) from err
   3624 except TypeError:
   3625     # If we have a listlike key, _check_indexing_error will raise
   3626     #  InvalidIndexError. Otherwise we fall through and re-raise
   3627     #  the TypeError.
   3628     self._check_indexing_error(key)

KeyError: 'Time (Quarter)'
```

In [ ]:
```
X_train, X_test, y_train, y_test = train_test_split(
...     X, y, test_size=0.33, random_state=42)
```

In [202…

```
mod = ar_select_order(diff4_wd, maxlag=10, seasonal=True, period=4,ic='aic')
mod.ar_lags
res = mod.model.fit()
print(res.summary())
```

```
                            AutoReg Model Results
==============================================================================
==
Dep. Variable:              TotalWine   No. Observations:
47
Model:                Seas. AutoReg(4)   Log Likelihood                  15.5
01
Method:                Conditional MLE   S.D. of innovations              0.1
69
Date:                Mon, 31 Oct 2022   AIC                            -13.0
02
Time:                        22:51:02   BIC                              2.8
49
Sample:                             4   HQIC                            -7.1
57
                                   47
==============================================================================
====
                 coef    std err          z      P>|z|      [0.025       0.
975]
------------------------------------------------------------------------------
----
const          0.1450      0.053      2.738      0.006       0.041        0
.249
s(2,4)        -0.0387      0.072     -0.534      0.593      -0.181        0
.103
s(3,4)        -0.0430      0.072     -0.597      0.551      -0.184        0
.098
s(4,4)        -0.0243      0.074     -0.328      0.743      -0.169        0
.121
TotalWine.L1   0.0024      0.116      0.020      0.984      -0.224        0
.229
TotalWine.L2   0.0512      0.115      0.444      0.657      -0.175        0
.277
TotalWine.L3   0.0525      0.115      0.456      0.649      -0.173        0
.278
TotalWine.L4  -0.6940      0.123     -5.644      0.000      -0.935       -0
.453
                                    Roots
==============================================================================
=
                  Real          Imaginary           Modulus         Frequenc
y
------------------------------------------------------------------------------
-
AR.1           -0.7680           -0.7637j            1.0830           -0.375
4
AR.2           -0.7680           +0.7637j            1.0830            0.375
```

```
4
AR.3            0.8058           -0.7610j            1.1084           -0.120
5
AR.4            0.8058           +0.7610j            1.1084            0.120
5
-----------------------------------------------------------------------------
-
```

```
/Users/dhavalgarg/opt/anaconda3/lib/python3.9/site-packages/statsmodels/tsa/
base/tsa_model.py:471: ValueWarning: An unsupported index was provided and w
ill be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
/Users/dhavalgarg/opt/anaconda3/lib/python3.9/site-packages/statsmodels/tsa/
base/tsa_model.py:471: ValueWarning: An unsupported index was provided and w
ill be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
```

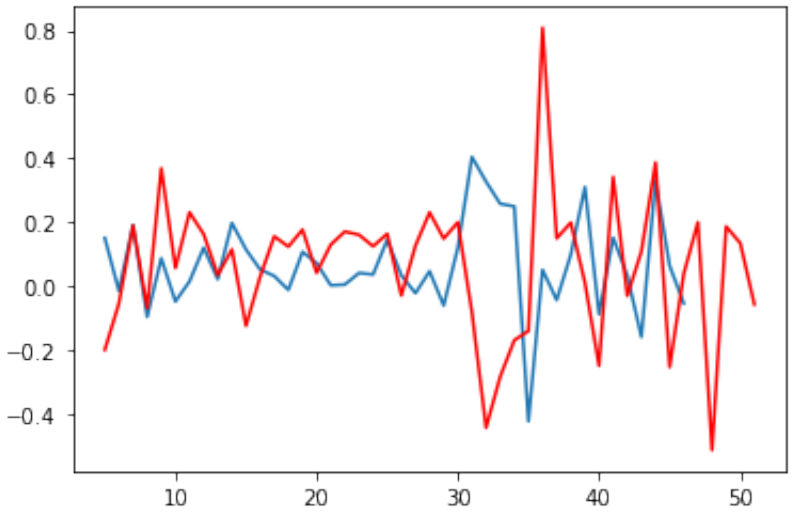
In [203…

```
out=res.predict()
out
```

Out[203]:
```
Time (Quarter)
5     0.151167
6    -0.014414
7     0.191458
8    -0.095833
9     0.087382
10   -0.047672
11    0.015060
12    0.120505
13    0.021976
14    0.197943
15    0.114882
16    0.052453
17    0.032075
18   -0.011544
19    0.107560
20    0.072339
21    0.002950
22    0.005185
23    0.041414
24    0.036758
25    0.145363
26    0.034114
27   -0.021697
28    0.046884
29   -0.060080
30    0.120796
31    0.404355
32    0.327316
33    0.258131
34    0.249786
35   -0.422611
36    0.051360
37   -0.043093
38    0.098781
39    0.310217
40   -0.087382
41    0.152157
42    0.033814
43   -0.158636
44    0.331710
45    0.063972
46   -0.053788
47         NaN
48         NaN
49         NaN
50         NaN
51         NaN
dtype: float64
```

In [217…
```
plt.plot(out)
plt.plot(diff4_wd,color='red')
```

Out[217]: [<matplotlib.lines.Line2D at 0x7fb9d54b1100>]



In [235…
```
from sklearn.metrics import mean_squared_error
```

In [238…
```
diff4_wd.dropna(inplace=True)
```

In [ ]:
```
mean_squared_error(diff4_wd['TotalWine'],out)
```

Train RMSE: 0.6589154504130144 Test RMSE: 0.8170648442164431

In [ ]: