DSC 275/475: Time Series Analysis and Forecasting (Fall 2021) Project-1

Question 1

In [2]:
```
import pandas as pd
import numpy as np
from matplotlib import pyplot
import matplotlib.pyplot as plt
```

In [115…
```
df=pd.read_csv("Problem1_DataSet.csv")
               #,header=0, index_col=0, parse_dates=True, squeeze=True)
df
```

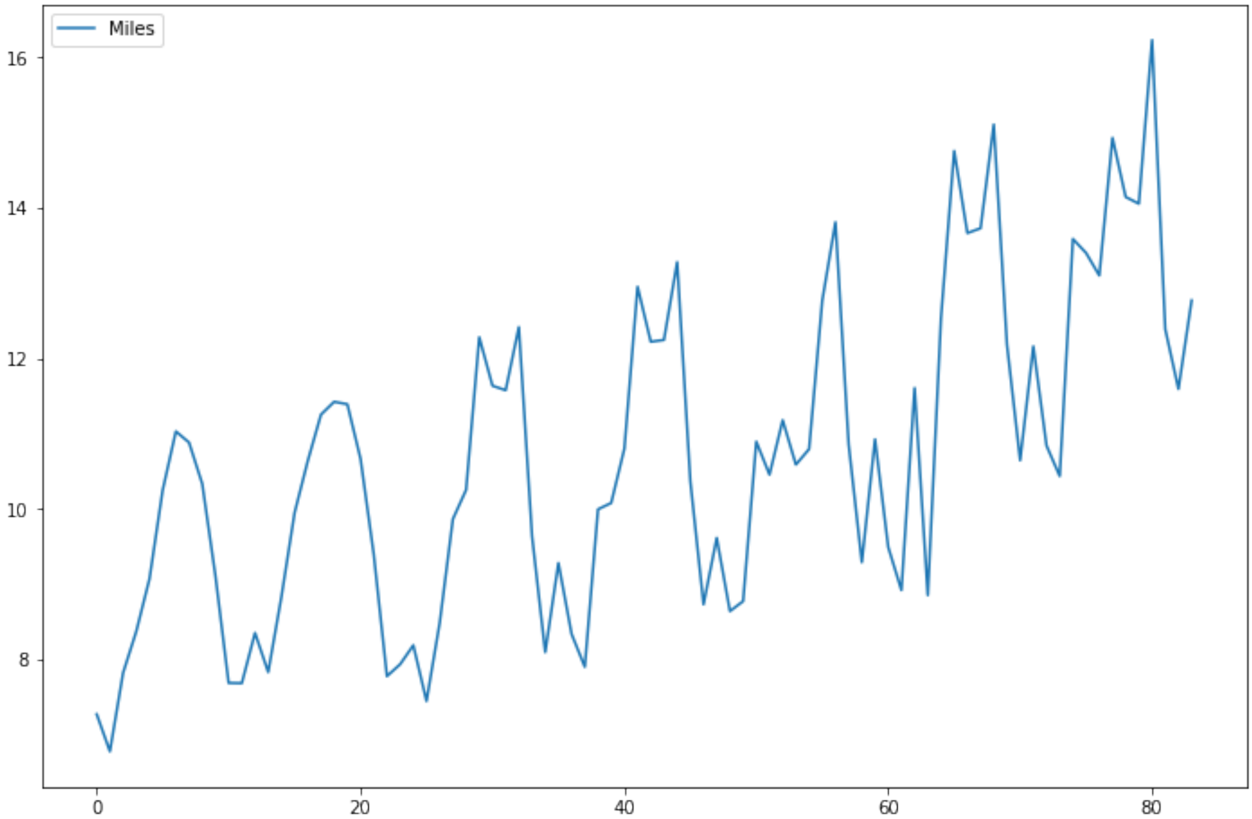
Out[115…

	Month	Miles
0	Jan-64	7.269
1	Feb-64	6.775
2	Mar-64	7.819
3	Apr-64	8.371
4	May-64	9.069
...	...	...
79	Aug-70	14.057
80	Sep-70	16.234
81	Oct-70	12.389
82	Nov-70	11.594
83	Dec-70	12.772

84 rows × 2 columns

In [4]:
```
x = df.Month
y = df.Miles
```

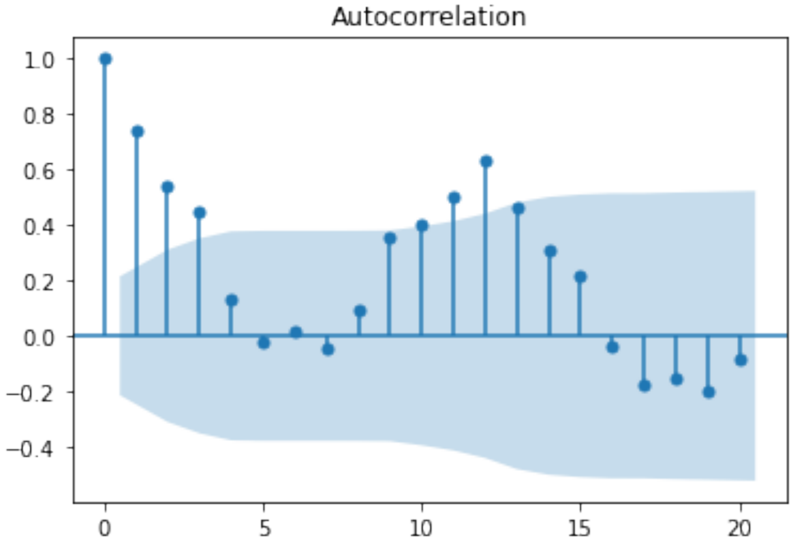
In [5]:
```
df.plot(figsize=(12,8))
pyplot.show()
```



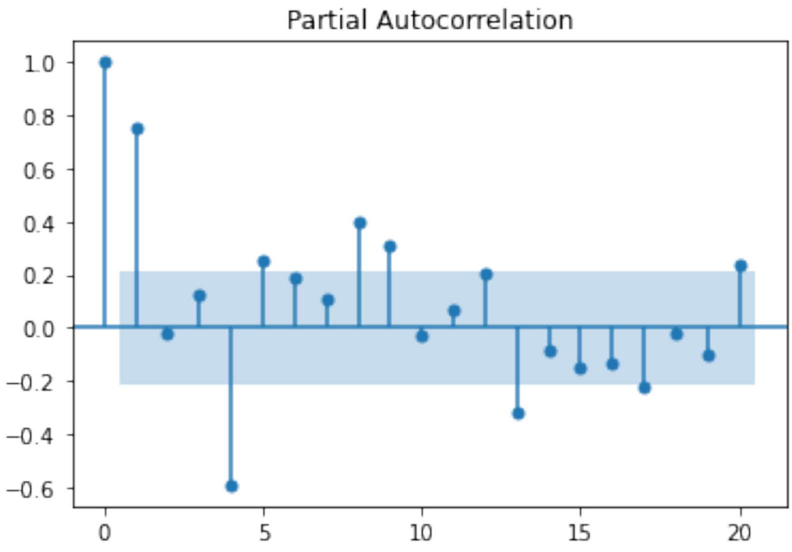
Question 2

In [6]:
```
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
```

In [7]:
```
figure= plot_acf(y)
```



In [8]:
```
figure= plot_pacf(y)
```

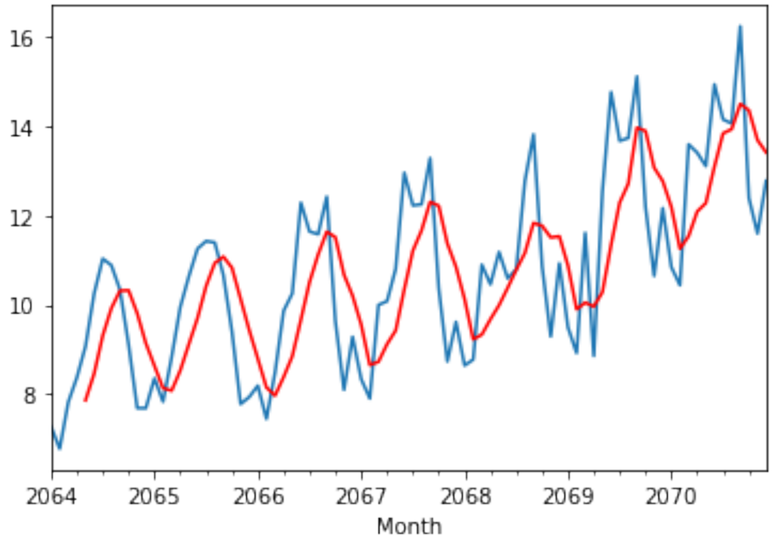


ACF shows an oscillation, indicating seasonality of 12.

Question 3:

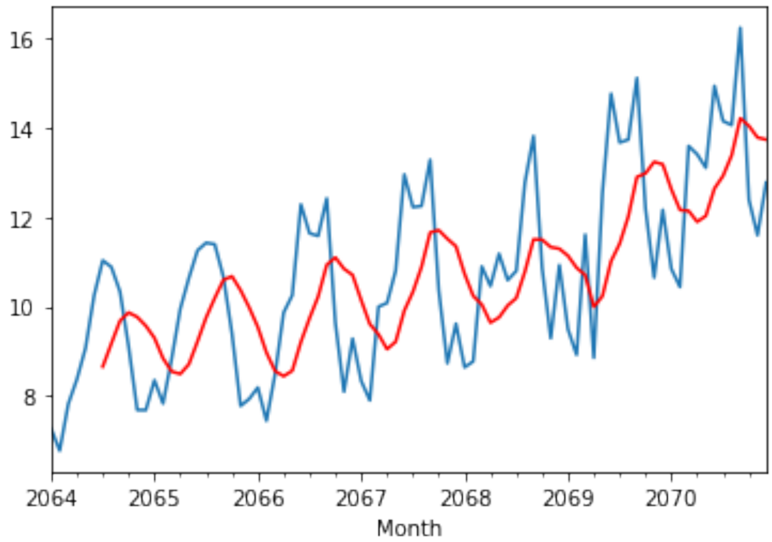
In [113…

```
rolling = df.rolling(window=5)
rolling_mean = rolling.mean()
# plot original and transformed dataset
df.plot()
rolling_mean.plot(color='red')
pyplot.show()
```



In [114…

```
rolling = df.rolling(window=7)
rolling_mean = rolling.mean()
# plot original and transformed dataset
df.plot()
rolling_mean.plot(color='red')
pyplot.show()
```

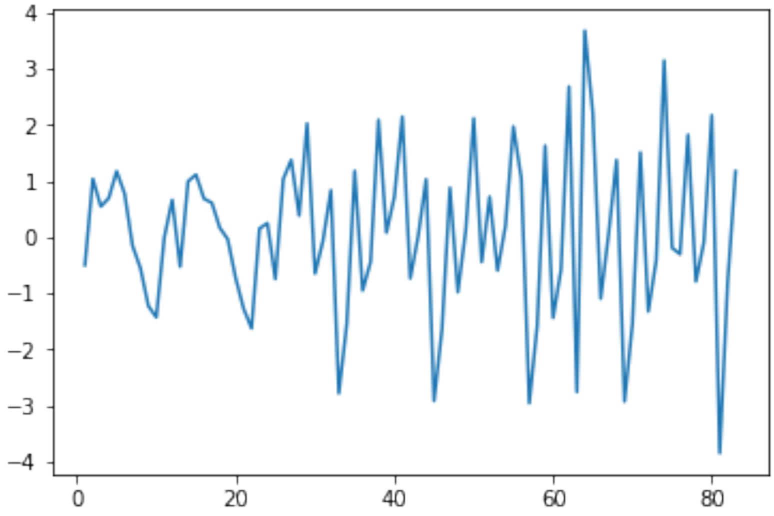


Suitable choice for the moving average window length can be 5,6 or 7

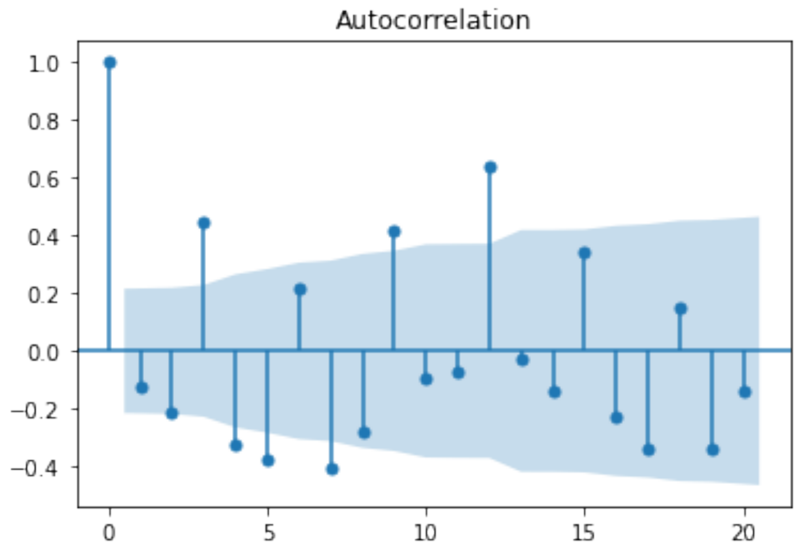
Question 4: The trend line from the graph is increasing

Question 5:

In [10]:
```
diff = y.diff()
diff = diff.iloc[1:]
plt.plot(diff)
plt.show()
```



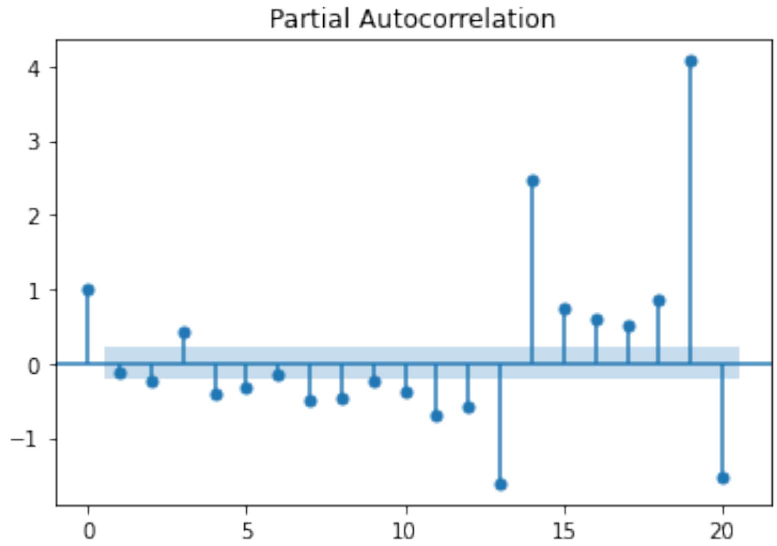
In [11]:
```
figure= plot_acf(diff)
```



The significant lags based on the ACF: 3,4,5,7,9,12

In [12]:
```
figure= plot_pacf(diff)
```

```
/opt/anaconda3/lib/python3.8/site-packages/statsmodels/regression/linear_mo
del.py:1434: RuntimeWarning: invalid value encountered in sqrt
  return rho, np.sqrt(sigmasq)
```



The significant lags based on the PACF: 2,3,4,5,7,8,11,20

Question 6:

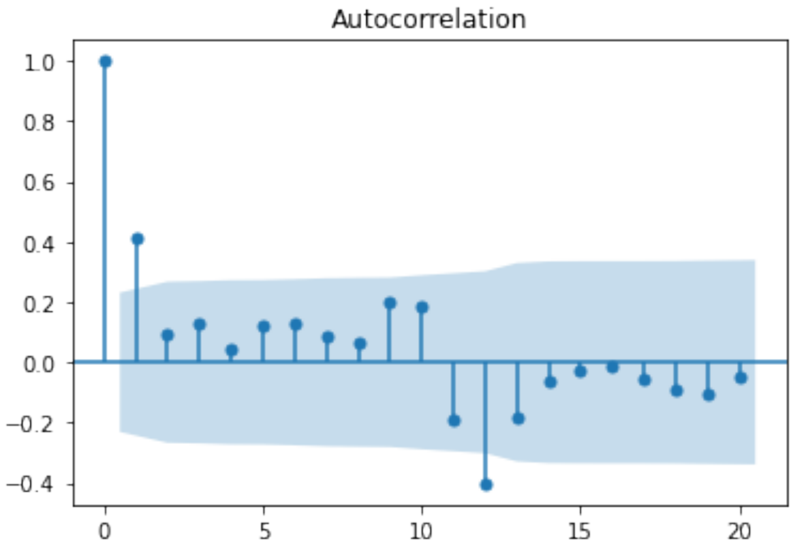
In [13]:
```
df['Seasonal First Difference']=df['Miles']-df['Miles'].shift(12)
```

In [14]:
```
a=df['Seasonal First Difference'].dropna()
```

In [15]:
```
a
```

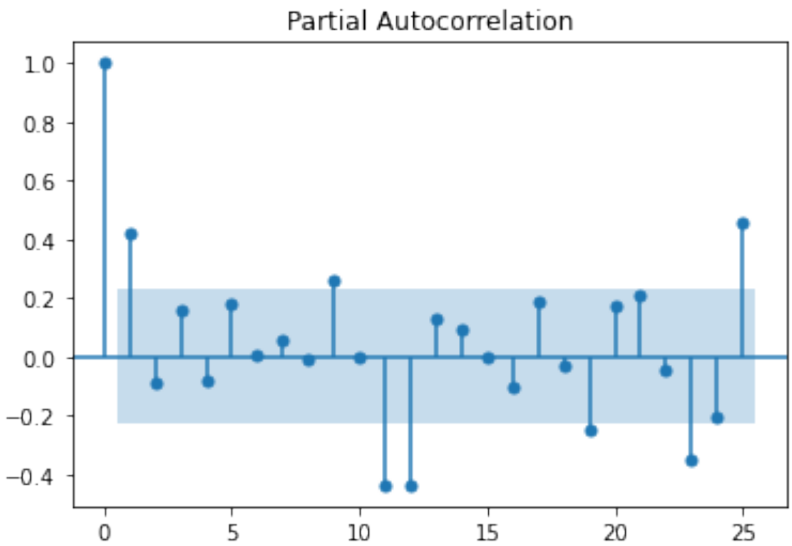
Out[15]:
```
12      1.081
13      1.054
14      1.010
15      1.577
16      1.569
        ...
79      0.326
80      1.124
81      0.204
82      0.949
83      0.611
Name: Seasonal First Difference, Length: 72, dtype: float64
```

In [16]:
```
figure= plot_acf(a, lags=20)
```



The significant lags based on the ACF: 1,12

In [17]:
```
figure= plot_pacf(a,lags=25)
```



The significant lags based on the PACF: 1,9,11,12,19,25

In [18]:
```
df[12:]
```

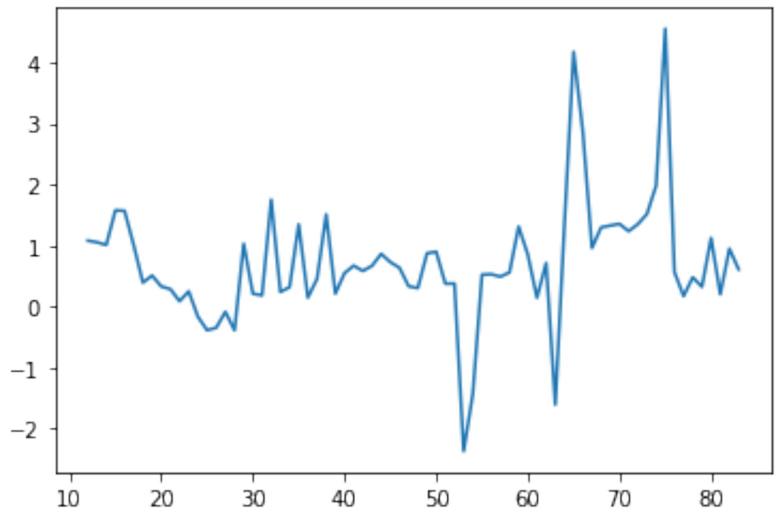
Out[18]:

	Month	Miles	Seasonal First Difference
12	Jan-65	8.350	1.081
13	Feb-65	7.829	1.054
14	Mar-65	8.829	1.010
15	Apr-65	9.948	1.577
16	May-65	10.638	1.569
...	...	...	...
79	Aug-70	14.057	0.326
80	Sep-70	16.234	1.124
81	Oct-70	12.389	0.204
82	Nov-70	11.594	0.949
83	Dec-70	12.772	0.611

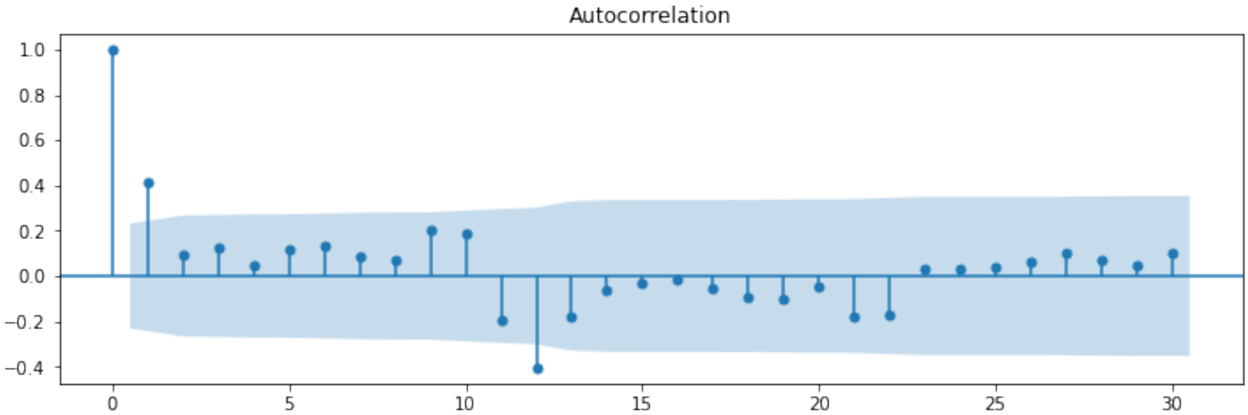
72 rows × 3 columns

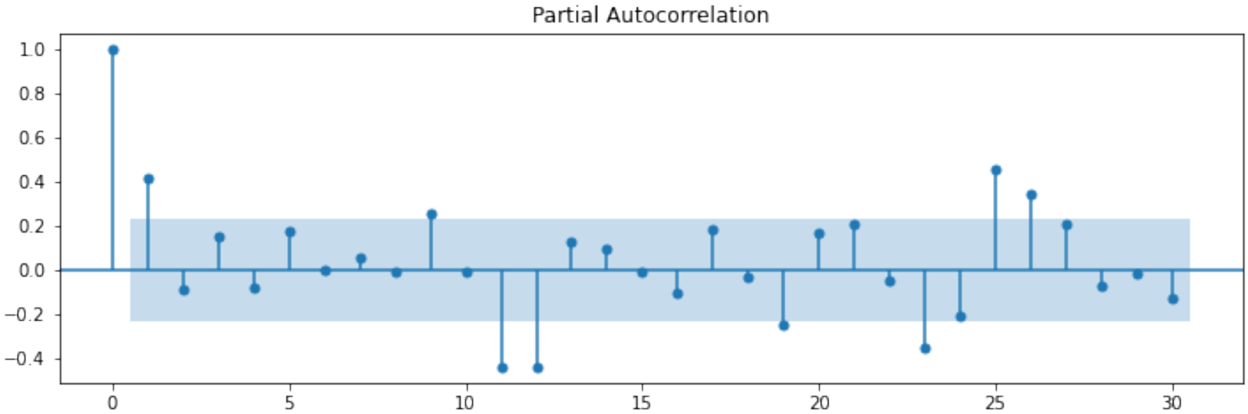
In [19]:
```
df['Seasonal First Difference'].plot()
```

Out[19]: <AxesSubplot:>



In [20]:
```
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(),lag
ax2 = fig.add_subplot(212)
fig = sm.graphics.tsa.plot_pacf(df['Seasonal First Difference'].dropna(),la
```





The significant lags based on the ACF: 1,12 The significant lags based on the PACF: 1,9,11,12,19,23,25,26

Question 7:

In [48]:
```
data= df[0:72] #first 6 years data only
```

In [49]:
```
from pmdarima import auto_arima
import warnings
warnings.filterwarnings("ignore")
```

In [50]:
```
# calculating best values
decomp= auto_arima(data['Miles'], start_p=0, start_q=0,
                  max_p=3,max_q=3,m=12,start_P=0, max_P=3,
                  start_Q=0, max_Q=3, d=None, D=None, trace=True,alpha=0.05
                  seasonal=True,stepwise=False)
```

```
ARIMA(0,0,0)(0,1,0)[12] intercept   : AIC=164.080, Time=0.01 sec
ARIMA(0,0,0)(0,1,1)[12] intercept   : AIC=153.783, Time=0.09 sec
ARIMA(0,0,0)(0,1,2)[12] intercept   : AIC=154.367, Time=0.15 sec
ARIMA(0,0,0)(0,1,3)[12] intercept   : AIC=156.234, Time=0.32 sec
ARIMA(0,0,0)(1,1,0)[12] intercept   : AIC=152.719, Time=0.04 sec
ARIMA(0,0,0)(1,1,1)[12] intercept   : AIC=154.373, Time=0.07 sec
ARIMA(0,0,0)(1,1,2)[12] intercept   : AIC=156.264, Time=0.35 sec
ARIMA(0,0,0)(1,1,3)[12] intercept   : AIC=158.216, Time=1.21 sec
ARIMA(0,0,0)(2,1,0)[12] intercept   : AIC=154.305, Time=0.10 sec
ARIMA(0,0,0)(2,1,1)[12] intercept   : AIC=156.218, Time=0.32 sec
ARIMA(0,0,0)(2,1,2)[12] intercept   : AIC=158.216, Time=0.68 sec
```

```
ARIMA(0,0,0)(2,1,3)[12] intercept   : AIC=160.216, Time=0.55 sec
ARIMA(0,0,0)(3,1,0)[12] intercept   : AIC=156.228, Time=0.32 sec
ARIMA(0,0,0)(3,1,1)[12] intercept   : AIC=158.216, Time=1.31 sec
ARIMA(0,0,0)(3,1,2)[12] intercept   : AIC=160.216, Time=0.64 sec
ARIMA(0,0,1)(0,1,0)[12] intercept   : AIC=148.341, Time=0.02 sec
ARIMA(0,0,1)(0,1,1)[12] intercept   : AIC=146.837, Time=0.05 sec
ARIMA(0,0,1)(0,1,2)[12] intercept   : AIC=148.556, Time=0.12 sec
ARIMA(0,0,1)(0,1,3)[12] intercept   : AIC=150.524, Time=0.46 sec
ARIMA(0,0,1)(1,1,0)[12] intercept   : AIC=146.584, Time=0.06 sec
ARIMA(0,0,1)(1,1,1)[12] intercept   : AIC=148.571, Time=0.10 sec
ARIMA(0,0,1)(1,1,2)[12] intercept   : AIC=150.544, Time=0.17 sec
ARIMA(0,0,1)(1,1,3)[12] intercept   : AIC=inf, Time=1.44 sec
ARIMA(0,0,1)(2,1,0)[12] intercept   : AIC=148.568, Time=0.19 sec
ARIMA(0,0,1)(2,1,1)[12] intercept   : AIC=inf, Time=0.66 sec
ARIMA(0,0,1)(2,1,2)[12] intercept   : AIC=152.366, Time=0.83 sec
ARIMA(0,0,1)(3,1,0)[12] intercept   : AIC=150.503, Time=0.39 sec
ARIMA(0,0,1)(3,1,1)[12] intercept   : AIC=inf, Time=1.91 sec
ARIMA(0,0,2)(0,1,0)[12] intercept   : AIC=150.207, Time=0.03 sec
ARIMA(0,0,2)(0,1,1)[12] intercept   : AIC=148.292, Time=0.07 sec
ARIMA(0,0,2)(0,1,2)[12] intercept   : AIC=150.079, Time=0.21 sec
ARIMA(0,0,2)(0,1,3)[12] intercept   : AIC=151.953, Time=0.50 sec
ARIMA(0,0,2)(1,1,0)[12] intercept   : AIC=148.058, Time=0.12 sec
ARIMA(0,0,2)(1,1,1)[12] intercept   : AIC=150.037, Time=0.18 sec
ARIMA(0,0,2)(1,1,2)[12] intercept   : AIC=152.035, Time=0.39 sec
ARIMA(0,0,2)(2,1,0)[12] intercept   : AIC=150.035, Time=0.21 sec
ARIMA(0,0,2)(2,1,1)[12] intercept   : AIC=151.993, Time=1.17 sec
ARIMA(0,0,2)(3,1,0)[12] intercept   : AIC=152.027, Time=0.62 sec
ARIMA(0,0,3)(0,1,0)[12] intercept   : AIC=151.828, Time=0.06 sec
ARIMA(0,0,3)(0,1,1)[12] intercept   : AIC=149.037, Time=0.09 sec
ARIMA(0,0,3)(0,1,2)[12] intercept   : AIC=150.787, Time=0.23 sec
ARIMA(0,0,3)(1,1,0)[12] intercept   : AIC=148.867, Time=0.08 sec
ARIMA(0,0,3)(1,1,1)[12] intercept   : AIC=150.758, Time=0.11 sec
ARIMA(0,0,3)(2,1,0)[12] intercept   : AIC=150.746, Time=0.19 sec
ARIMA(1,0,0)(0,1,0)[12] intercept   : AIC=153.472, Time=0.02 sec
ARIMA(1,0,0)(0,1,1)[12] intercept   : AIC=149.196, Time=0.06 sec
ARIMA(1,0,0)(0,1,2)[12] intercept   : AIC=150.259, Time=0.16 sec
ARIMA(1,0,0)(0,1,3)[12] intercept   : AIC=152.259, Time=0.43 sec
ARIMA(1,0,0)(1,1,0)[12] intercept   : AIC=148.559, Time=0.10 sec
ARIMA(1,0,0)(1,1,1)[12] intercept   : AIC=150.497, Time=0.11 sec
ARIMA(1,0,0)(1,1,2)[12] intercept   : AIC=152.259, Time=0.29 sec
ARIMA(1,0,0)(1,1,3)[12] intercept   : AIC=inf, Time=1.69 sec
ARIMA(1,0,0)(2,1,0)[12] intercept   : AIC=150.466, Time=0.18 sec
ARIMA(1,0,0)(2,1,1)[12] intercept   : AIC=inf, Time=0.83 sec
ARIMA(1,0,0)(2,1,2)[12] intercept   : AIC=154.132, Time=0.69 sec
ARIMA(1,0,0)(3,1,0)[12] intercept   : AIC=152.154, Time=0.78 sec
ARIMA(1,0,0)(3,1,1)[12] intercept   : AIC=154.130, Time=1.18 sec
ARIMA(1,0,1)(0,1,0)[12] intercept   : AIC=150.181, Time=0.07 sec
ARIMA(1,0,1)(0,1,1)[12] intercept   : AIC=148.046, Time=0.10 sec
ARIMA(1,0,1)(0,1,2)[12] intercept   : AIC=149.816, Time=0.36 sec
ARIMA(1,0,1)(0,1,3)[12] intercept   : AIC=151.660, Time=0.43 sec
ARIMA(1,0,1)(1,1,0)[12] intercept   : AIC=147.792, Time=0.10 sec
ARIMA(1,0,1)(1,1,1)[12] intercept   : AIC=149.760, Time=0.16 sec
ARIMA(1,0,1)(1,1,2)[12] intercept   : AIC=151.760, Time=0.39 sec
ARIMA(1,0,1)(2,1,0)[12] intercept   : AIC=149.759, Time=0.21 sec
ARIMA(1,0,1)(2,1,1)[12] intercept   : AIC=151.709, Time=1.44 sec
ARIMA(1,0,1)(3,1,0)[12] intercept   : AIC=151.754, Time=0.71 sec
ARIMA(1,0,2)(0,1,0)[12] intercept   : AIC=152.295, Time=0.25 sec
ARIMA(1,0,2)(0,1,1)[12] intercept   : AIC=149.985, Time=0.19 sec
ARIMA(1,0,2)(0,1,2)[12] intercept   : AIC=151.730, Time=0.82 sec
ARIMA(1,0,2)(1,1,0)[12] intercept   : AIC=149.715, Time=0.25 sec
ARIMA(1,0,2)(1,1,1)[12] intercept   : AIC=151.684, Time=0.34 sec
```

```
 ARIMA(1,0,2)(2,1,0)[12] intercept   : AIC=151.681, Time=0.47 sec
 ARIMA(1,0,3)(0,1,0)[12] intercept   : AIC=153.570, Time=0.10 sec
 ARIMA(1,0,3)(0,1,1)[12] intercept   : AIC=151.008, Time=0.20 sec
 ARIMA(1,0,3)(1,1,0)[12] intercept   : AIC=150.866, Time=0.18 sec
 ARIMA(2,0,0)(0,1,0)[12] intercept   : AIC=154.062, Time=0.06 sec
 ARIMA(2,0,0)(0,1,1)[12] intercept   : AIC=150.812, Time=0.07 sec
 ARIMA(2,0,0)(0,1,2)[12] intercept   : AIC=151.995, Time=0.17 sec
 ARIMA(2,0,0)(0,1,3)[12] intercept   : AIC=153.993, Time=0.34 sec
 ARIMA(2,0,0)(1,1,0)[12] intercept   : AIC=150.205, Time=0.08 sec
 ARIMA(2,0,0)(1,1,1)[12] intercept   : AIC=152.178, Time=0.18 sec
 ARIMA(2,0,0)(1,1,2)[12] intercept   : AIC=153.994, Time=0.62 sec
 ARIMA(2,0,0)(2,1,0)[12] intercept   : AIC=152.165, Time=0.18 sec
 ARIMA(2,0,0)(2,1,1)[12] intercept   : AIC=inf, Time=1.32 sec
 ARIMA(2,0,0)(3,1,0)[12] intercept   : AIC=153.907, Time=0.97 sec
 ARIMA(2,0,1)(0,1,0)[12] intercept   : AIC=152.049, Time=0.10 sec
 ARIMA(2,0,1)(0,1,1)[12] intercept   : AIC=149.878, Time=0.32 sec
 ARIMA(2,0,1)(0,1,2)[12] intercept   : AIC=151.591, Time=1.04 sec
 ARIMA(2,0,1)(1,1,0)[12] intercept   : AIC=149.596, Time=0.31 sec
 ARIMA(2,0,1)(1,1,1)[12] intercept   : AIC=151.565, Time=0.31 sec
 ARIMA(2,0,1)(2,1,0)[12] intercept   : AIC=151.560, Time=0.71 sec
 ARIMA(2,0,2)(0,1,0)[12] intercept   : AIC=153.662, Time=0.21 sec
 ARIMA(2,0,2)(0,1,1)[12] intercept   : AIC=151.288, Time=0.46 sec
 ARIMA(2,0,2)(1,1,0)[12] intercept   : AIC=151.132, Time=0.44 sec
 ARIMA(2,0,3)(0,1,0)[12] intercept   : AIC=155.565, Time=0.28 sec
 ARIMA(3,0,0)(0,1,0)[12] intercept   : AIC=153.579, Time=0.07 sec
 ARIMA(3,0,0)(0,1,1)[12] intercept   : AIC=150.354, Time=0.09 sec
 ARIMA(3,0,0)(0,1,2)[12] intercept   : AIC=152.158, Time=0.24 sec
 ARIMA(3,0,0)(1,1,0)[12] intercept   : AIC=150.368, Time=0.11 sec
 ARIMA(3,0,0)(1,1,1)[12] intercept   : AIC=152.161, Time=0.19 sec
 ARIMA(3,0,0)(2,1,0)[12] intercept   : AIC=152.127, Time=0.29 sec
 ARIMA(3,0,1)(0,1,0)[12] intercept   : AIC=153.149, Time=0.06 sec
 ARIMA(3,0,1)(0,1,1)[12] intercept   : AIC=150.188, Time=0.14 sec
 ARIMA(3,0,1)(1,1,0)[12] intercept   : AIC=150.338, Time=0.19 sec
 ARIMA(3,0,2)(0,1,0)[12] intercept   : AIC=inf, Time=0.26 sec

Best model:  ARIMA(0,0,1)(1,1,0)[12] intercept
Total fit time: 39.987 seconds
```

A couple of approaches could be: Model 1: ARIMA(3,0,1)(1,0,0)[12] intercept : AIC=198.558, Time=0.26 sec Model 2: ARIMA(3,2,1)(3,2,1)[12] intercept : AIC=164.584, Time=0.10 sec Model 3: ARIMA(0,0,1)(1,1,0)[12] intercept : AIC=146.584, Time=0.07 sec

I'm using AIC as the evaluation criteria!

APPROACH 1: AIC value is 205.999 which is pretty decent, and the plot also looks very similar but it doesn’t look like it takes the trend into consideration because the plot is on the same linear plane. Rating of the pdq= 3/5

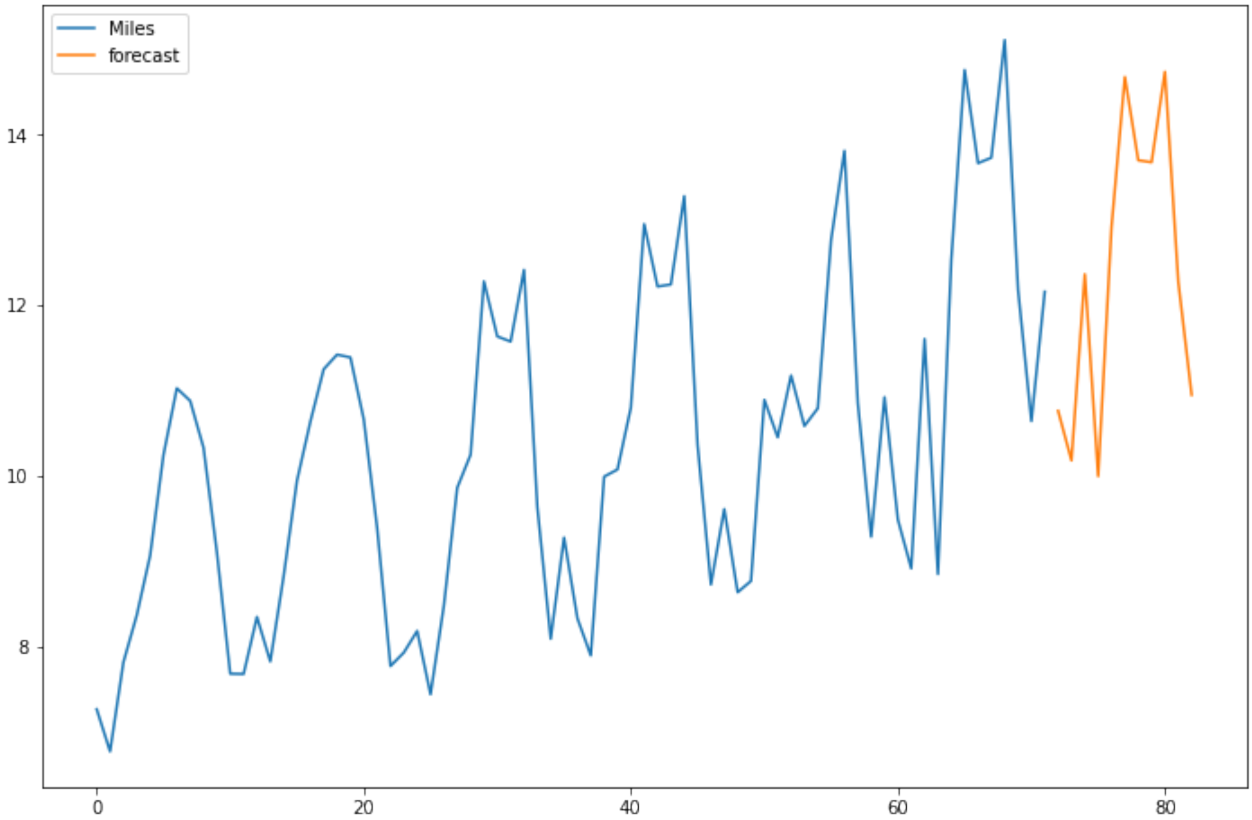
In [59]:

```
model=sm.tsa.statespace.SARIMAX(data['Miles'],order=(3, 0, 1),seasonal_orde
results=model.fit()
```

In [60]:

```
data['forecast'] = results.predict(start = 72, end = 84, dynamic= True)
data[['Miles', 'forecast']].plot(figsize=(12, 8))
```

Out[60]: <AxesSubplot:>



In [61]:
```
print(results.summary())
```

```
                                      SARIMAX Results
============================================================================
================
Dep. Variable:                                 Miles   No. Observations:
83
Model:             SARIMAX(3, 0, 1)x(1, 0, [], 12)   Log Likelihood
-97.000
Date:                               Tue, 26 Oct 2021   AIC
205.999
Time:                                       01:21:30   BIC
220.512
Sample:                                            0   HQIC
211.830
                                                - 83
Covariance Type:                                 opg
============================================================================
===
                 coef    std err          z      P>|z|      [0.025      0.9
75]
----------------------------------------------------------------------------
---
ar.L1          0.3417      0.203      1.681      0.093      -0.057       0.
740
ar.L2          0.2035      0.229      0.890      0.374      -0.245       0.
652
ar.L3          0.3960      0.127      3.120      0.002       0.147       0.
645
ma.L1          0.4981      0.236      2.109      0.035       0.035       0.
961
ar.S.L12       0.8168      0.095      8.557      0.000       0.630       1.
004
sigma2         0.6785      0.074      9.145      0.000       0.533       0.
824
============================================================================
========
Ljung-Box (L1) (Q):                   0.06   Jarque-Bera (JB):
64.62
Prob(Q):                              0.81   Prob(JB):
0.00
Heteroskedasticity (H):               4.31   Skew:
0.27
Prob(H) (two-sided):                  0.00   Kurtosis:
7.29
============================================================================
========

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

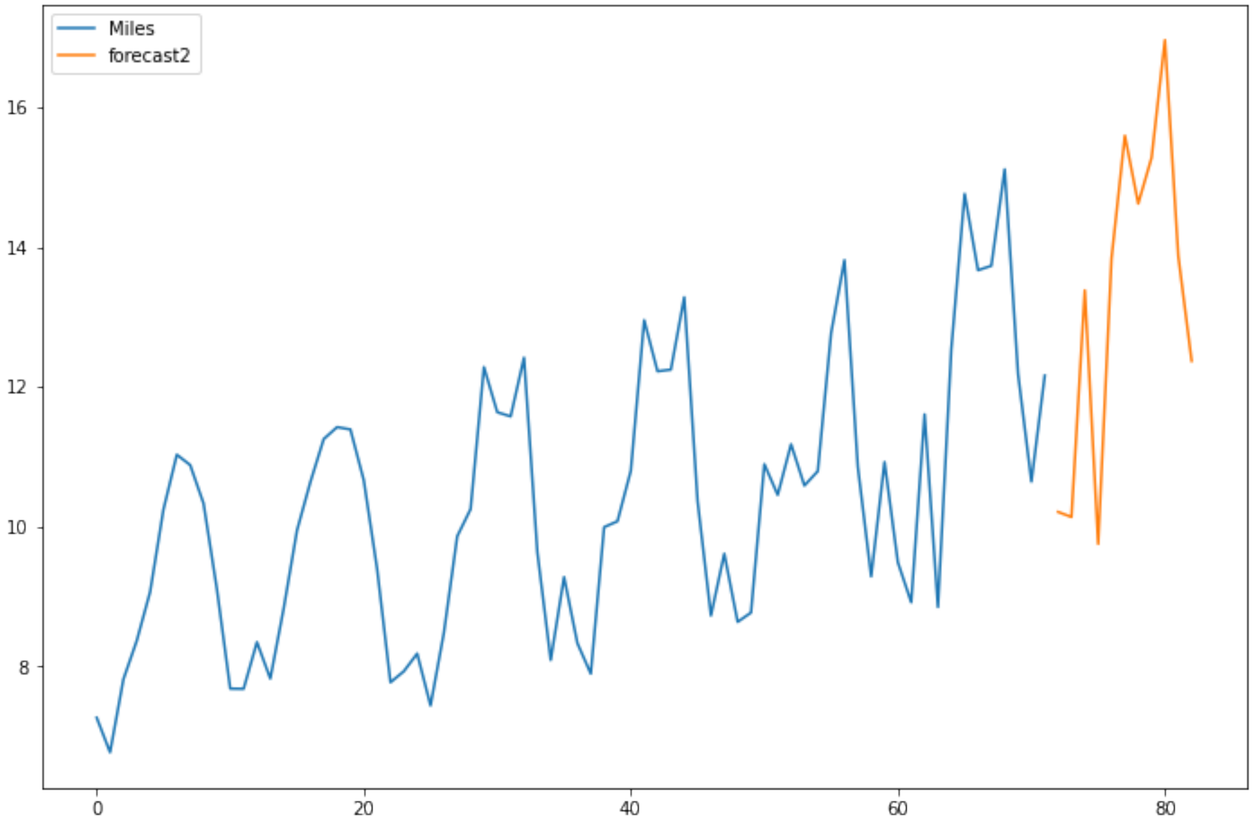
In [54]:
```
#IGNORE
data=data.append(pd.Series(), ignore_index=True)
data=data.append(pd.Series(), ignore_index=True)
data=data.append(pd.Series(), ignore_index=True)
data=data.append(pd.Series(), ignore_index=True)
data=data.append(pd.Series(), ignore_index=True)
data=data.append(pd.Series(), ignore_index=True)
data=data.append(pd.Series(), ignore_index=True)
data=data.append(pd.Series(), ignore_index=True)
data=data.append(pd.Series(), ignore_index=True)
data=data.append(pd.Series(), ignore_index=True)
data=data.append(pd.Series(), ignore_index=True)
```

APPROACH 2: AIC value is 164.374 which is much better than 1, and the plot also looks VERY similar and looks like it takes the trend into consideration because the plot is slightly higher but again it's not exactly perfect when it comes to the seasonal pattern. Rating of the pdq= 4/5

In [58]:
```
model2=sm.tsa.statespace.SARIMAX(data['Miles'],order=(3, 2, 1),seasonal_ord
results2=model2.fit()
```

In [62]:
```
data['forecast2'] = results2.predict(start = 72, end = 84, dynamic= True)
data[['Miles', 'forecast2']].plot(figsize=(12, 8))
```

Out[62]: <AxesSubplot:>



In [63]:

```
print(results2.summary())
```

```
                                     SARIMAX Results                                     
=========================================================================================
Dep. Variable:                                 Miles   No. Observations:                83
Model:             SARIMAX(3, 2, 1)x(3, 2, 1, 12)   Log Likelihood              -73.187
Date:                            Tue, 26 Oct 2021   AIC                         164.374
Time:                                    01:21:41   BIC                         182.762
Sample:                                         0   HQIC                        171.520
                                             - 83                                         
Covariance Type:                              opg                                         
===================================================================================
                   coef    std err          z      P>|z|      [0.025      0.975]
-----------------------------------------------------------------------------------
ar.L1           -0.2153      0.182     -1.185      0.236      -0.571       0.141
ar.L2           -0.3698      0.206     -1.799      0.072      -0.773       0.033
ar.L3           -0.1302      0.267     -0.488      0.625      -0.653       0.392
ma.L1           -0.9968      6.409     -0.156      0.876     -13.558      11.565
ar.S.L12        -0.2898    201.242     -0.001      0.999    -394.717     394.137
ar.S.L24        -0.0011    173.101  -6.58e-06      1.000    -339.274     339.271
ar.S.L36         0.1662     88.785      0.002      0.999    -173.850     174.182
ma.S.L12        -0.6282    209.806     -0.003      0.998    -411.840     410.584
sigma2           0.9646     10.108      0.095      0.924     -18.848      20.777
========================================================================================
Ljung-Box (L1) (Q):                   0.19   Jarque-Bera (JB):                   30.17
Prob(Q):                              0.66   Prob(JB):                            0.00
Heteroskedasticity (H):               1.71   Skew:                               -0.46
Prob(H) (two-sided):                  0.25   Kurtosis:                            6.44
========================================================================================

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

APPROACH 3:AIC value is 160.908 which is much better, and the plot also looks VERY similar but again looks like it does not take the trend into consideration because the plot is on the same height. Rating of the pdq= 3/5

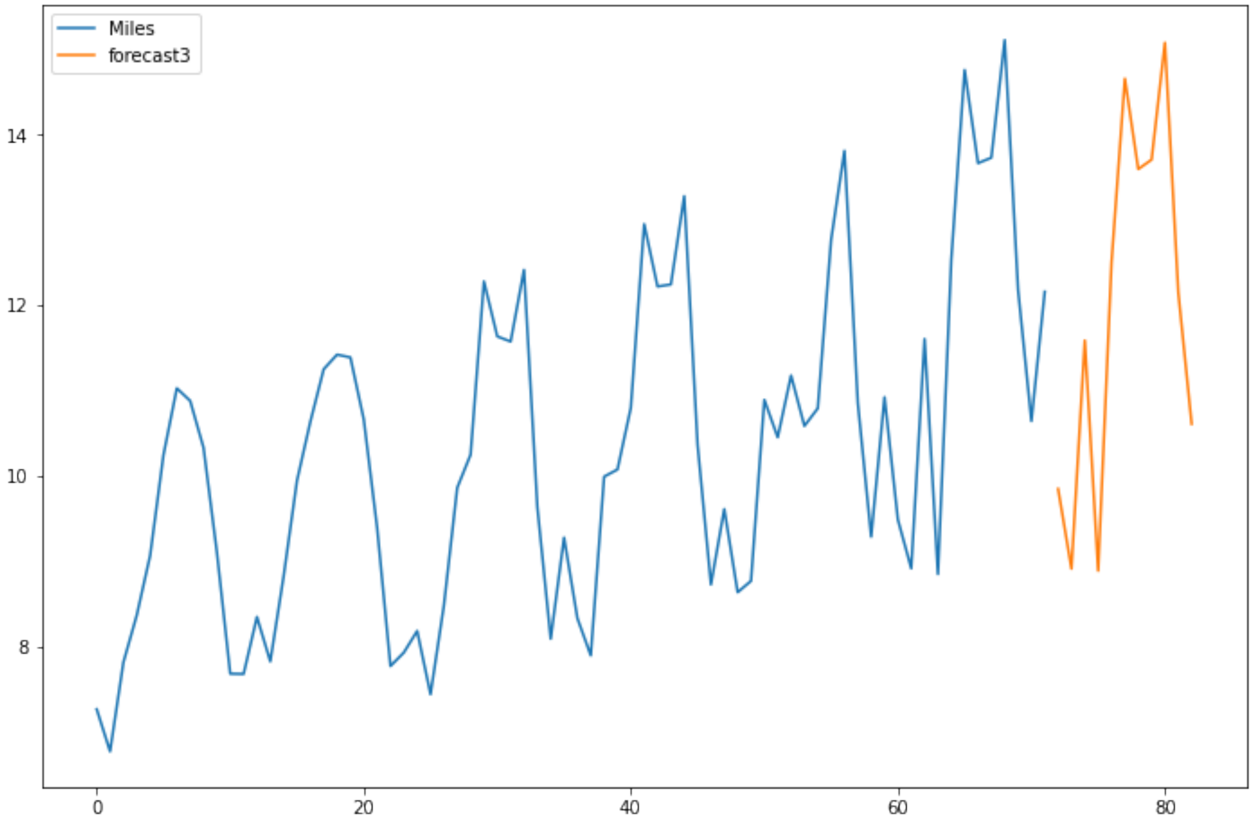
In [64]:

```
model3=sm.tsa.statespace.SARIMAX(data['Miles'],order=(0, 0, 1),seasonal_or
results3=model3.fit()
```

In [65]:

```
data['forecast3'] = results3.predict(start = 72, end = 84, dynamic= True)
data[['Miles', 'forecast3']].plot(figsize=(12, 8))
```

Out[65]: <AxesSubplot:>



In [66]:

```
print(results3.summary())
```

```
                                     SARIMAX Results
==========================================================================================
Dep. Variable:                              Miles   No. Observations:                   83
Model:             SARIMAX(0, 0, 1)x(1, 1, [], 12)   Log Likelihood                 -77.454
Date:                            Tue, 26 Oct 2021   AIC                            160.908
Time:                                    01:21:55   BIC                            167.696
Sample:                                         0   HQIC                           163.607
                                             - 83
Covariance Type:                              opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ma.L1          0.6822      0.117      5.814      0.000       0.452       0.912
ar.S.L12      -0.0242      0.256     -0.095      0.925      -0.527       0.478
sigma2         0.7659      0.128      5.981      0.000       0.515       1.017
===================================================================================
Ljung-Box (L1) (Q):                   0.29   Jarque-Bera (JB):                67.33
Prob(Q):                              0.59   Prob(JB):                         0.00
Heteroskedasticity (H):               2.22   Skew:                            -0.50
Prob(H) (two-sided):                  0.06   Kurtosis:                         7.67
===================================================================================

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

APPROACH 4: AIC value is 159.498 which is much better, and the plot also looks similar but again it does not follow seasonality pattern properly. Rating of the pdq= 3/5

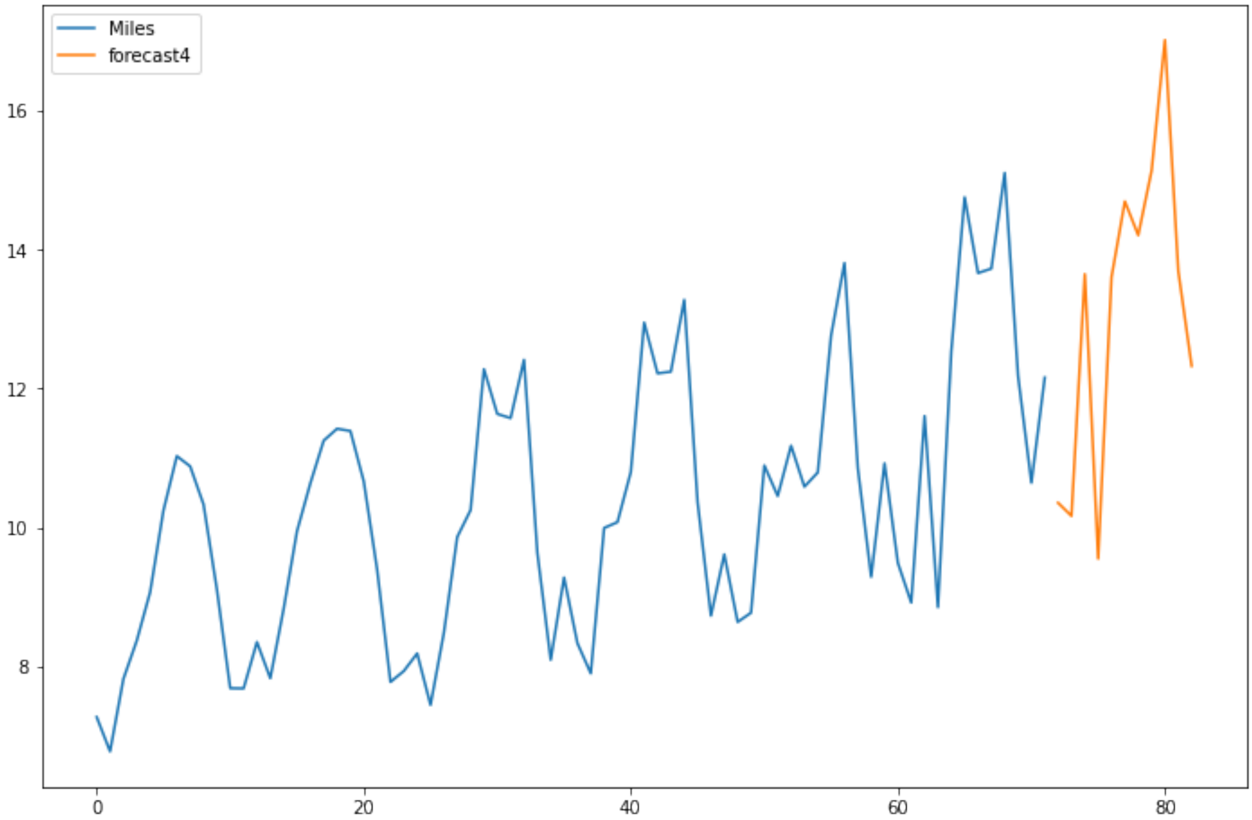
In [68]:

```python
model4=sm.tsa.statespace.SARIMAX(data['Miles'],order=(3, 1, 2),seasonal_or
results4=model4.fit()
```

In [69]:

```python
data['forecast4'] = results4.predict(start = 72, end = 84, dynamic= True)
data[['Miles', 'forecast4']].plot(figsize=(12, 8))
```

Out[69]: <AxesSubplot:>



In [70]:

```
print(results4.summary())
```

```
                                      SARIMAX Results
==========================================================================
===============
Dep. Variable:                                  Miles   No. Observations:
83
Model:             SARIMAX(3, 1, 2)x(3, 2, 2, 12)   Log Likelihood
-68.749
Date:                            Tue, 26 Oct 2021   AIC
159.498
Time:                                    01:22:09   BIC
182.163
Sample:                                         0   HQIC
168.327
                                             - 83
Covariance Type:                              opg
==========================================================================
===
                 coef    std err          z      P>|z|      [0.025      0.9
75]
--------------------------------------------------------------------------
---
ar.L1         -0.5792     10.878     -0.053      0.958     -21.900      20.
742
ar.L2          0.1435      4.572      0.031      0.975      -8.817       9.
104
ar.L3         -0.2707      2.814     -0.096      0.923      -5.786       5.
245
ma.L1          0.2175      4.450      0.049      0.961      -8.504       8.
939
ma.L2         -0.7506      3.537     -0.212      0.832      -7.683       6.
181
ar.S.L12      -0.5515   7380.969  -7.47e-05      1.000   -1.45e+04    1.45e
+04
ar.S.L24      -0.6778   5373.108     -0.000      1.000   -1.05e+04    1.05e
+04
ar.S.L36      -0.2563   2371.795     -0.000      1.000   -4648.890    4648.
377
ma.S.L12      -0.4878   9679.525  -5.04e-05      1.000    -1.9e+04     1.9e
+04
ma.S.L24       0.8435   6818.636      0.000      1.000   -1.34e+04    1.34e
+04
sigma2         0.6422   3255.233      0.000      1.000   -6379.497    6380.
781
==========================================================================
========
Ljung-Box (L1) (Q):                  0.06   Jarque-Bera (JB):
74.27
Prob(Q):                             0.80   Prob(JB):
0.00
Heteroskedasticity (H):              1.44   Skew:
-0.76
Prob(H) (two-sided):                 0.44   Kurtosis:
8.33
==========================================================================
========

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

APPROACH 5: AIC value is 159.498 which is much better, and the plot also looks similar but again it does not follow seasonality pattern properly. Rating of the pdq= 3/5

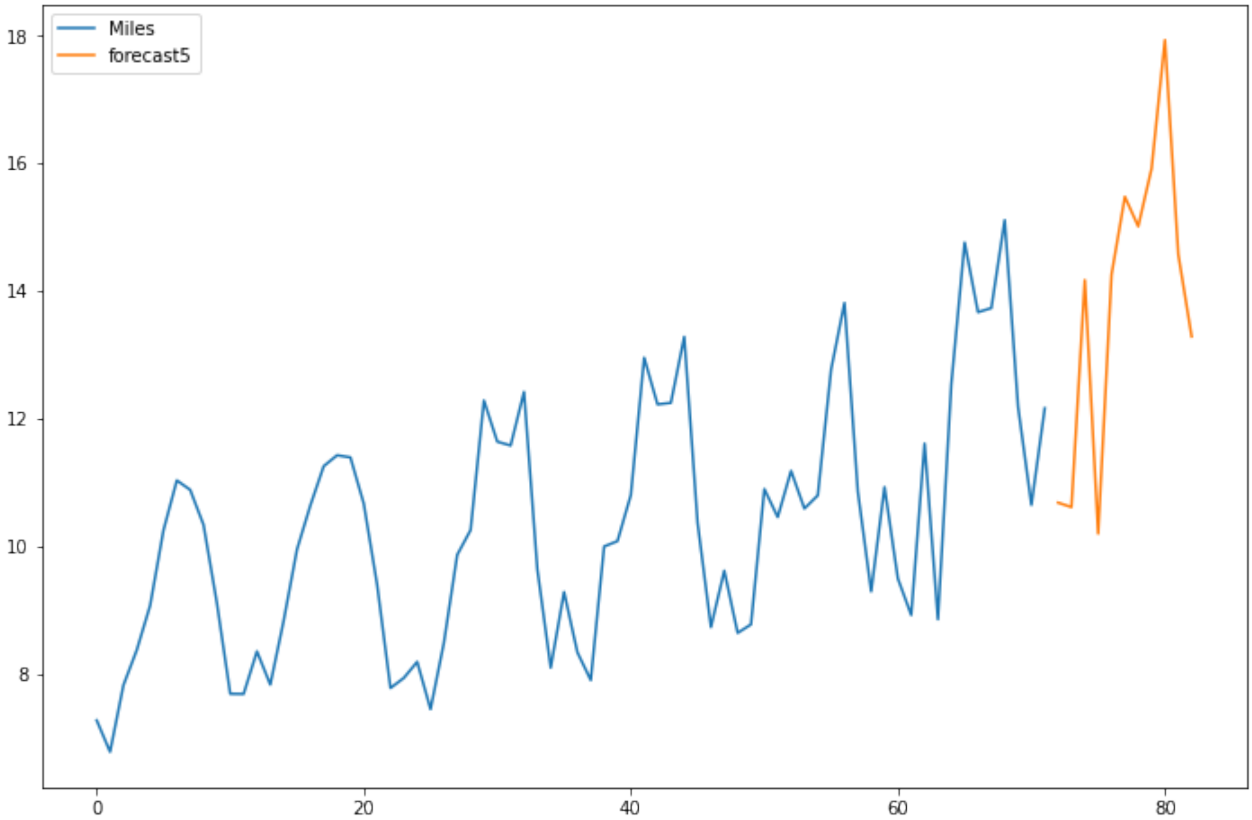
In [71]:

```
model5=sm.tsa.statespace.SARIMAX(data['Miles'],order=(3, 2, 3),seasonal_or
results5=model5.fit()
```

In [72]:

```
data['forecast5'] = results5.predict(start = 72, end = 84, dynamic= True)
data[['Miles', 'forecast5']].plot(figsize=(12, 8))
```

Out[72]: <AxesSubplot:>



In [73]:

```
print(results4.summary())
```

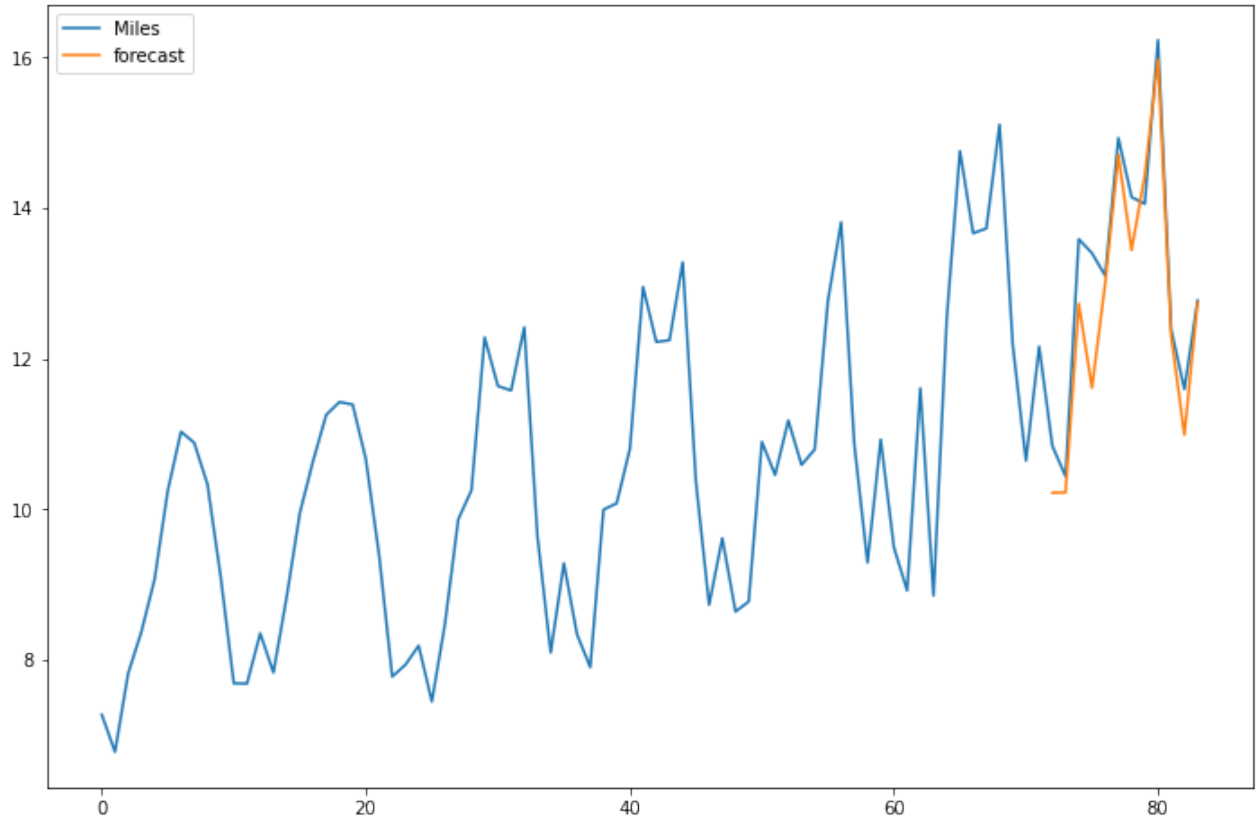
```
                                 SARIMAX Results
================================================================================
===============
Dep. Variable:                                 Miles   No. Observations:
83
Model:             SARIMAX(3, 1, 2)x(3, 2, 2, 12)   Log Likelihood
-68.749
Date:                            Tue, 26 Oct 2021   AIC
159.498
Time:                                    01:22:24   BIC
182.163
Sample:                                         0   HQIC
168.327
                                             - 83
Covariance Type:                              opg
================================================================================
===
                 coef    std err          z      P>|z|      [0.025      0.9
75]
--------------------------------------------------------------------------------
---
ar.L1         -0.5792     10.878     -0.053      0.958     -21.900      20.
742
ar.L2          0.1435      4.572      0.031      0.975      -8.817       9.
104
ar.L3         -0.2707      2.814     -0.096      0.923      -5.786       5.
245
ma.L1          0.2175      4.450      0.049      0.961      -8.504       8.
939
ma.L2         -0.7506      3.537     -0.212      0.832      -7.683       6.
181
ar.S.L12      -0.5515   7380.969  -7.47e-05      1.000   -1.45e+04    1.45e
+04
ar.S.L24      -0.6778   5373.108     -0.000      1.000   -1.05e+04    1.05e
+04
ar.S.L36      -0.2563   2371.795     -0.000      1.000   -4648.890    4648.
377
ma.S.L12      -0.4878   9679.525  -5.04e-05      1.000    -1.9e+04     1.9e
+04
ma.S.L24       0.8435   6818.636      0.000      1.000   -1.34e+04    1.34e
+04
sigma2         0.6422   3255.233      0.000      1.000   -6379.497    6380.
781
================================================================================
========
Ljung-Box (L1) (Q):                   0.06   Jarque-Bera (JB):
74.27
Prob(Q):                              0.80   Prob(JB):
0.00
Heteroskedasticity (H):               1.44   Skew:
-0.76
Prob(H) (two-sided):                  0.44   Kurtosis:
8.33
================================================================================
========

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

Question 8:

In [74]:
```
modelfinal=sm.tsa.statespace.SARIMAX(df['Miles'],order=(3, 1, 3),seasonal_(
resultsfinal=modelfinal.fit()
```

In [75]:
```
df['forecast']=resultsfinal.predict(start=72,end=84,dynamic=True)
df[['Miles','forecast']].plot(figsize=(12,8))
```

Out[75]: <AxesSubplot:>



In [76]:
```
print(resultsfinal.summary())
```

```
                                     SARIMAX Results
==========================================================================================
Dep. Variable:                              Miles   No. Observations:
84
Model:             SARIMAX(3, 1, 3)x(3, 1, [], 12)   Log Likelihood
-80.303
Date:                            Tue, 26 Oct 2021   AIC
180.607
Time:                                    01:22:38   BIC
203.233
Sample:                                         0   HQIC
189.605
                                             - 84
Covariance Type:                              opg
==============================================================================
===
                 coef    std err          z      P>|z|      [0.025      0.9
75]
------------------------------------------------------------------------------
---
ar.L1         -0.6678      0.309     -2.158      0.031      -1.274      -0.
061
ar.L2         -0.6677      0.316     -2.113      0.035      -1.287      -0.
048
ar.L3          0.2879      0.260      1.106      0.269      -0.222       0.
798
ma.L1          0.0303      0.293      0.104      0.918      -0.543       0.
604
ma.L2          0.0988      0.173      0.571      0.568      -0.240       0.
438
ma.L3         -0.6635      0.160     -4.141      0.000      -0.978      -0.
349
ar.S.L12      -0.8695      0.136     -6.386      0.000      -1.136      -0.
603
ar.S.L24      -0.5246      0.303     -1.734      0.083      -1.118       0.
068
ar.S.L36      -0.1831      0.446     -0.411      0.681      -1.057       0.
690
sigma2         0.4869      0.069      7.073      0.000       0.352       0.
622
==============================================================================
========
Ljung-Box (L1) (Q):                  0.00   Jarque-Bera (JB):
64.06
Prob(Q):                             0.97   Prob(JB):
0.00
Heteroskedasticity (H):              2.99   Skew:
-1.07
Prob(H) (two-sided):                 0.01   Kurtosis:
7.14
==============================================================================
========

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

In [81]:

```
df[72:] #compare actual values to forecasted values!
```

Out[81]:

	Month	Miles	Seasonal First Difference	forecast
72	Jan-70	10.840	1.349	10.217580
73	Feb-70	10.436	1.517	10.222063
74	Mar-70	13.589	1.982	12.731674
75	Apr-70	13.402	4.550	11.612775
76	May-70	13.103	0.566	12.998826
77	Jun-70	14.933	0.174	14.710351
78	Jul-70	14.147	0.480	13.443494
79	Aug-70	14.057	0.326	14.433915
80	Sep-70	16.234	1.124	15.979700
81	Oct-70	12.389	0.204	12.300071
82	Nov-70	11.594	0.949	10.988532
83	Dec-70	12.772	0.611	12.748958

In [98]:

```
maerror= mean_absolute_error((df["Miles"]).iloc[72:83], df["forecast"].iloc
mae
```

Out[98]: 0.13139640993766555

USED: order=(3, 1, 3),seasonal_order=(3,1,0,12) The forecast is very close to the actual values as you can see in the graph and the table, but not exactly the same. I tried playing around with different p,q, P, Q values to see which one would follow the actual values closely, turns out this is the best one I could find. The AIC value is 180.607 and the forecast nearly follows the actual values.
It also follows the increasing trend of actual values. Mean absolute error is also 0.13139!

Problem 2:

In [99]:

```
df1=pd.read_csv("TotalWine.csv")
```

In [100…

```
df1.head()
```

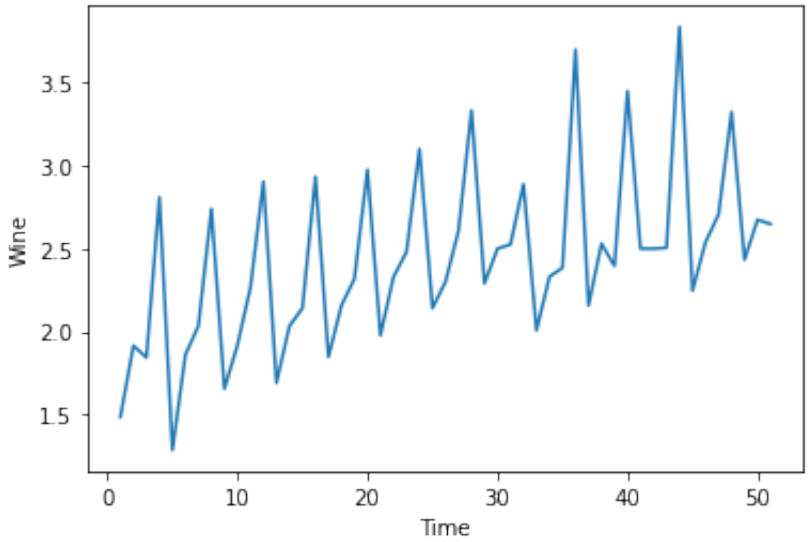
Out[100…

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

Part A

In [101…

```
x1 = df1["Time"]
y1 = df1.TotalWine
plt.plot (x1, y1)
plt.xlabel ('Time')
plt.ylabel ('Wine')
plt.show()
```



Part B

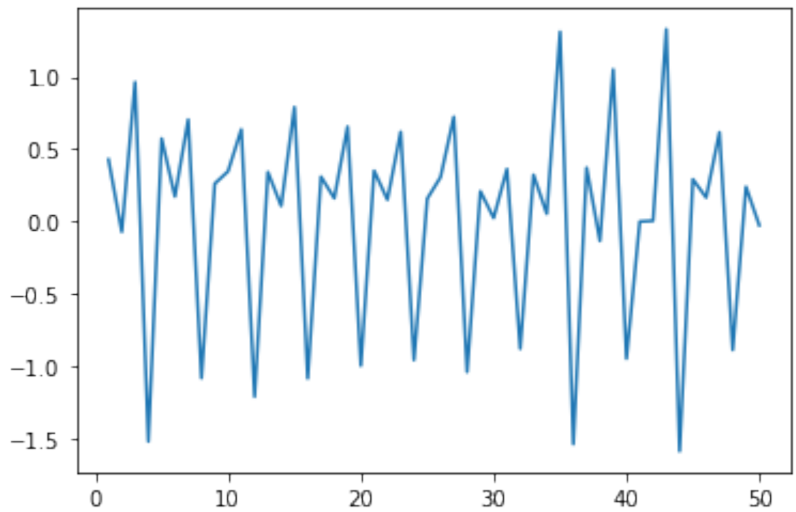
In [102…

```
df1['Seasonal First Difference 1']=y1-y1.shift(1)
df1['Seasonal First Difference 2']=y1-y1.shift(2)
df1['Seasonal First Difference 4']=y1-y1.shift(4)
df1['Seasonal First Difference 6']=y1-y1.shift(6)
```

In [103…

```
df1['Seasonal First Difference 1'].plot()
```

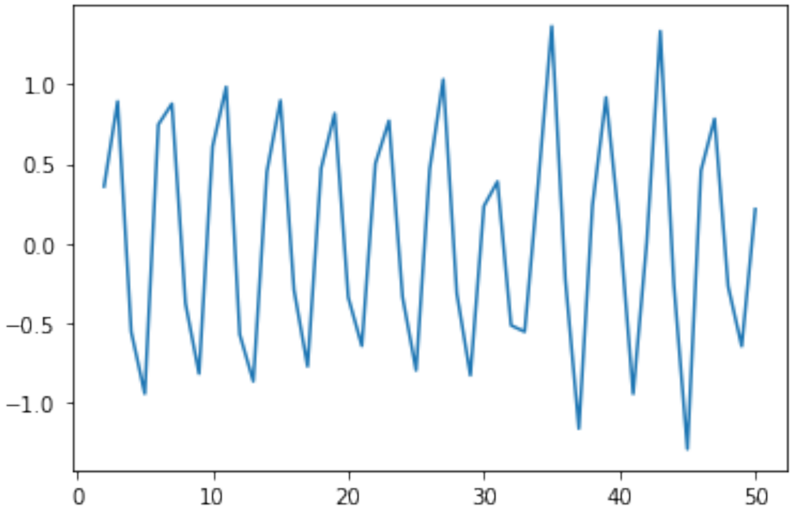
Out[103… <AxesSubplot:>



In [104…

```
df1['Seasonal First Difference 2'].plot()
```

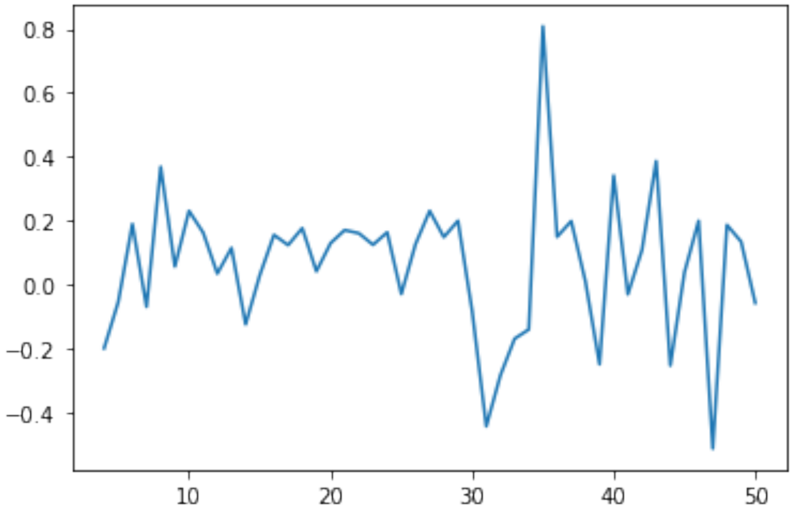
Out[104… <AxesSubplot:>



In [105…

```
df1['Seasonal First Difference 4'].plot()
```

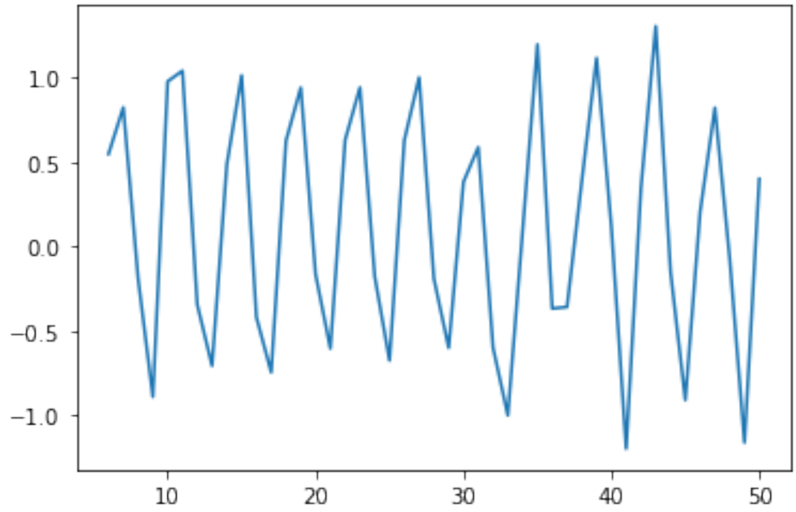
Out[105… <AxesSubplot:>



In [106…

```
df1['Seasonal First Difference 6'].plot()
```

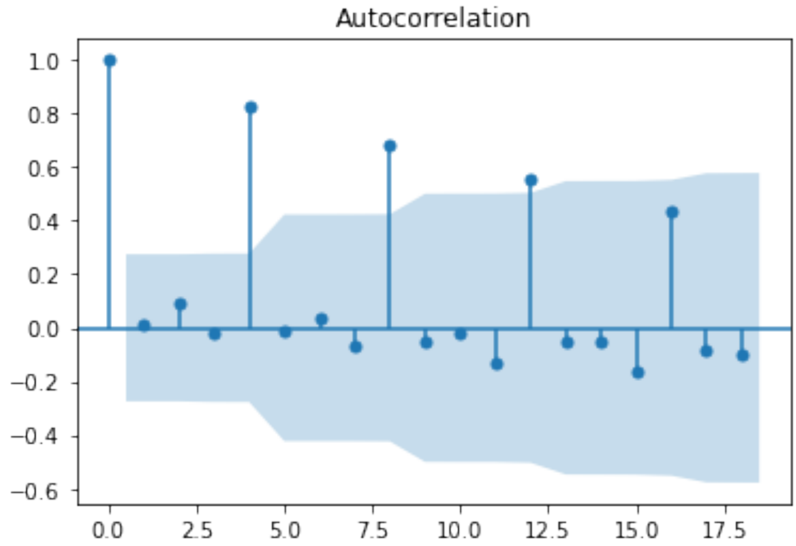
Out[106… <AxesSubplot:>



4 difference is best suited!

In [107…

```
figure= plot_acf(y1)
```



Seasonality period is 4-lag long.

In [108…

```
from statsmodels.tsa.api import AR
bestlag= AR(df1['Seasonal First Difference 4'].dropna().values).select_orde
bestlag
```

Out[108… 5

Best lag is 5,we should use AR(5)

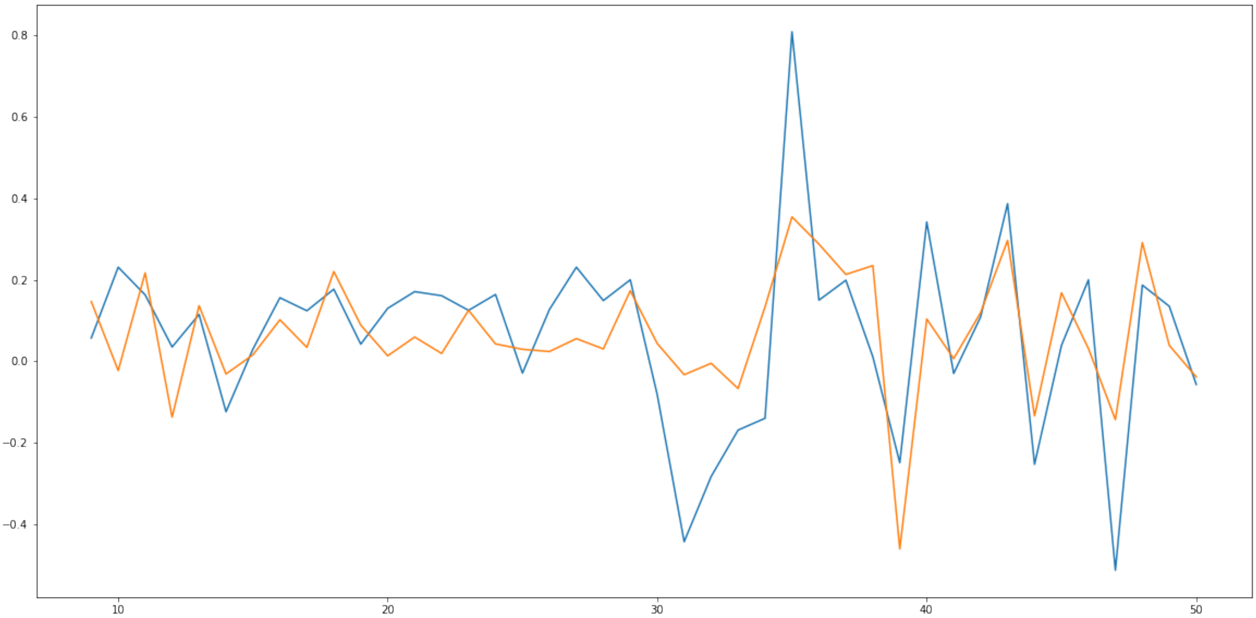
In [109…

```
model= AR(df1['Seasonal First Difference 4'].dropna().values).fit(maxlag=5
forecast= model.predict()
```

In [110…

```
plt.figure(figsize=(20,10))
df1['Seasonal First Difference 4'].dropna().iloc[5:].plot()
plt.plot(df1['Seasonal First Difference 4'].dropna().iloc[5:].index, foreca
```

Out[110… [<matplotlib.lines.Line2D at 0x166864280>]



In [111…

```
from sklearn.metrics import mean_absolute_error
mae= mean_absolute_error(df1['Seasonal First Difference 4'].dropna().iloc[5
mae
```

Out[111… 0.13139640993766555

In [ ]: