Dscc275 Project_01

October 26, 2021

In [1]:
```
# Problem1
```

In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib.pyplot import MultipleLocator
from statsmodels.graphics.tsaplots import plot_acf,plot_pacf
import statsmodels.api as sm
```

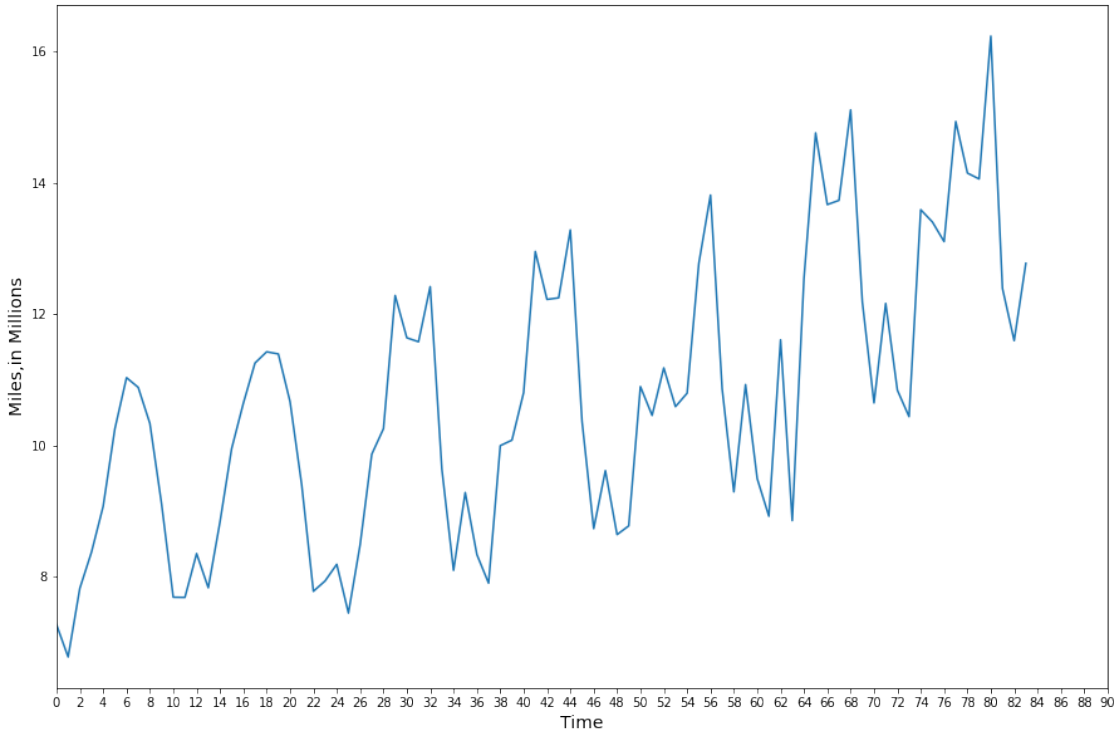
In [3]:
```
# (1)
```

In [4]:
```
df = pd.read_csv("Problem1_DataSet.csv")
df.head(12)
```

Out[4]:

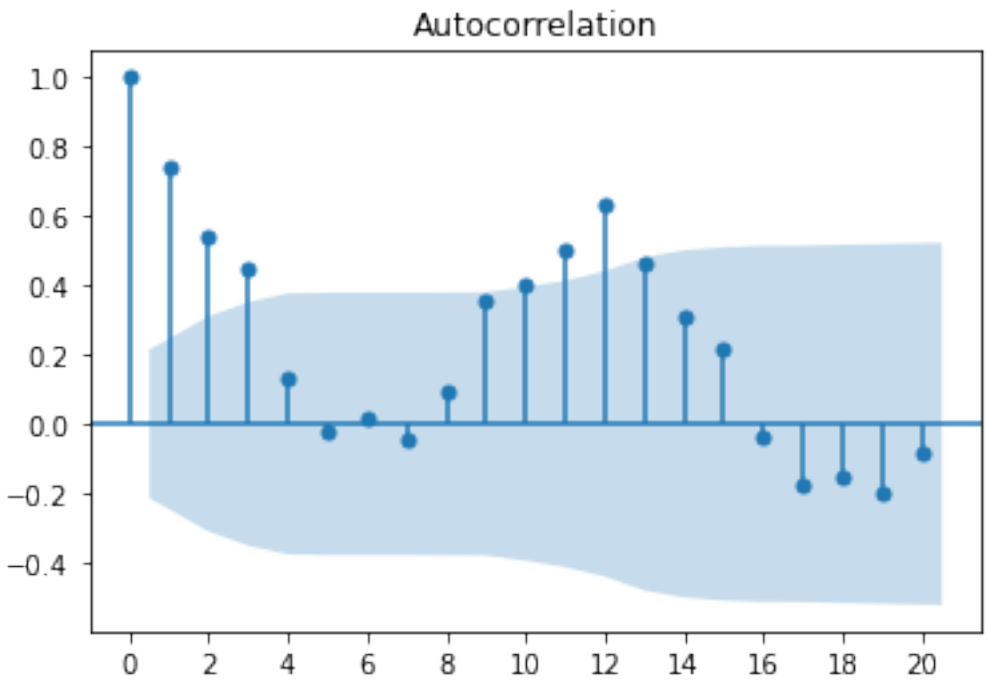
	Month	Miles, in Millions
0	Jan-1964	7.269
1	Feb-1964	6.775
2	Mar-1964	7.819
3	Apr-1964	8.371
4	May-1964	9.069
5	Jun-1964	10.248
6	Jul-1964	11.030
7	Aug-1964	10.882
8	Sep-1964	10.333
9	Oct-1964	9.109
10	Nov-1964	7.685
11	Dec-1964	7.682

In [5]:
```
plt.figure(figsize=[15,10])
plt.plot(df['Miles, in Millions'])
plt.xlabel('Time',fontsize=14)
plt.ylabel('Miles,in Millions',fontsize=14)
x_major_locator=MultipleLocator(2)
ax=plt.gca()
ax.xaxis.set_major_locator(x_major_locator)
plt.xlim(0,90)
plt.show()
```



In [6]:
```
# (2)
```

In [7]:
```
ACF = plot_acf(df['Miles, in Millions'])
x_major_locator=MultipleLocator(2)
ax=ACF.gca()
ax.xaxis.set_major_locator(x_major_locator)
ACF.show()
```



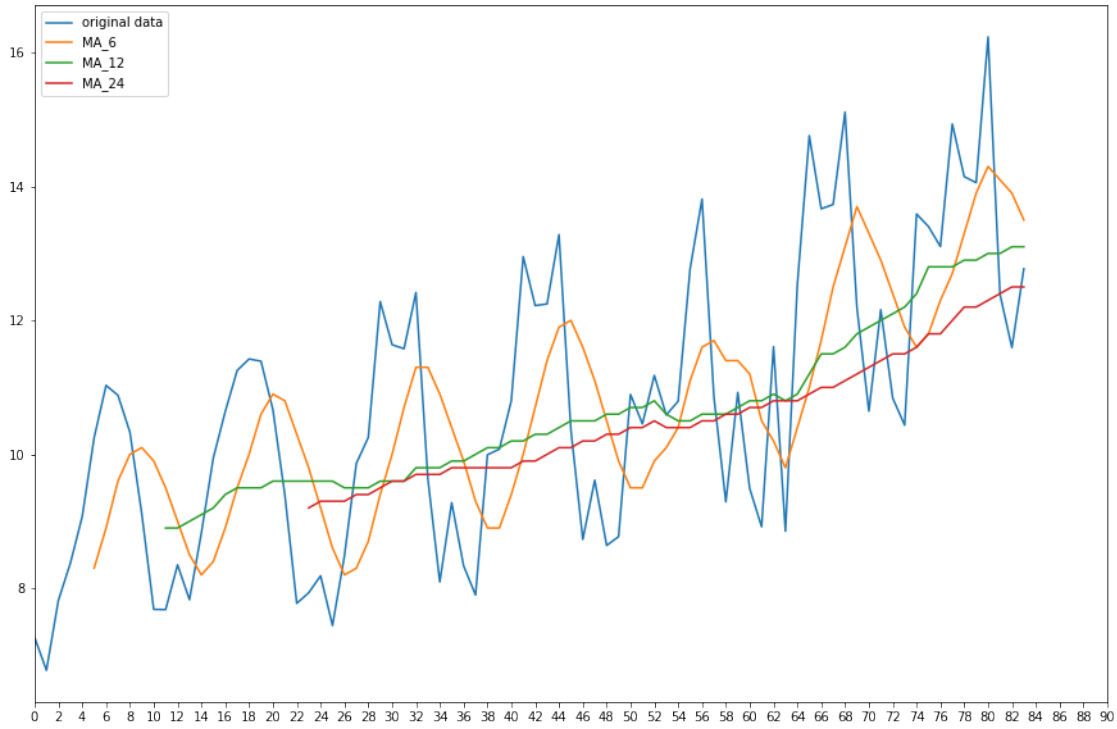

```
In [8]: # The seasonal period is 12.

In [9]: # (3)

In [10]: MA_6 = np.round(df['Miles, in Millions'].rolling(6).mean(),1)
         df.insert(df.shape[1],'MA_6',MA_6)
         MA_12 = np.round(df['Miles, in Millions'].rolling(12).mean(),1)
         df.insert(df.shape[1],'MA_12',MA_12)
         MA_24 = np.round(df['Miles, in Millions'].rolling(24).mean(),1)
         df.insert(df.shape[1],'MA_24',MA_24)
         df
```

Out[10]:

	Month	Miles, in Millions	MA_6	MA_12	MA_24
0	Jan-1964	7.269	NaN	NaN	NaN
1	Feb-1964	6.775	NaN	NaN	NaN
2	Mar-1964	7.819	NaN	NaN	NaN
3	Apr-1964	8.371	NaN	NaN	NaN
4	May-1964	9.069	NaN	NaN	NaN
..	...	...	...	...	...
79	Aug-1970	14.057	13.9	12.9	12.2
80	Sep-1970	16.234	14.3	13.0	12.3
81	Oct-1970	12.389	14.1	13.0	12.4
82	Nov-1970	11.594	13.9	13.1	12.5
83	Dec-1970	12.772	13.5	13.1	12.5

[84 rows x 5 columns]

In [11]:
```
plt.figure(figsize=[15,10])
plt.plot(df['Miles, in Millions'], label='original data')
plt.plot(df['MA_6'],label='MA_6')
plt.plot(df['MA_12'],label='MA_12')
plt.plot(df['MA_24'],label='MA_24')
plt.legend(loc=2)
x_major_locator=MultipleLocator(2)
ax=plt.gca()
ax.xaxis.set_major_locator(x_major_locator)
plt.xlim(0,90)
plt.show()
```



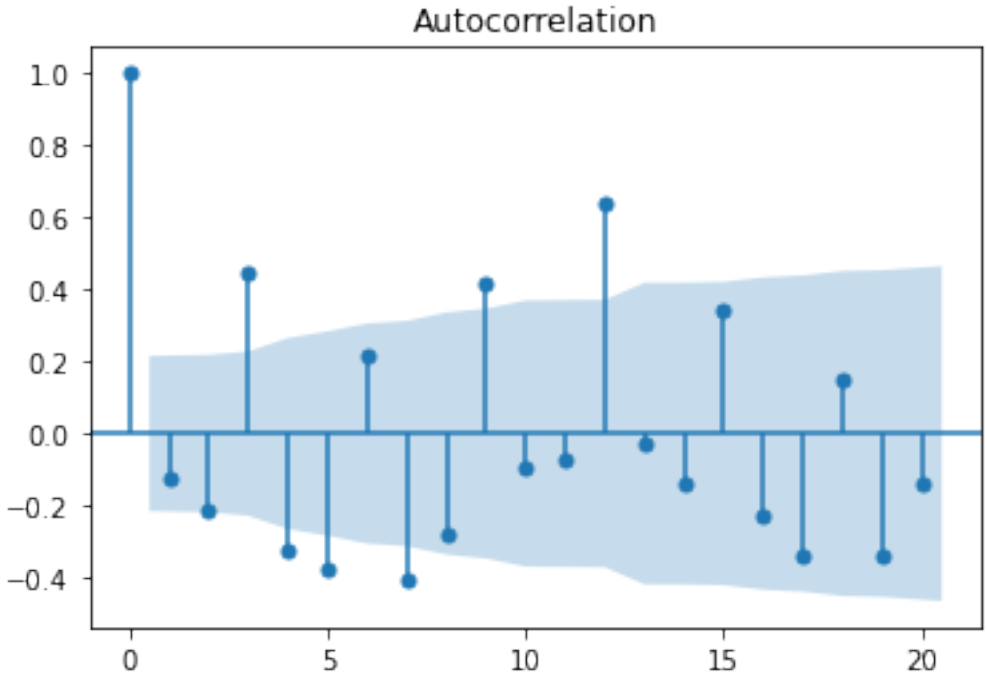
In [12]:
```
# I think 12 or 24 are suitable, because they are more smoothing.
# window length = 6 still has seasonal factor.
```

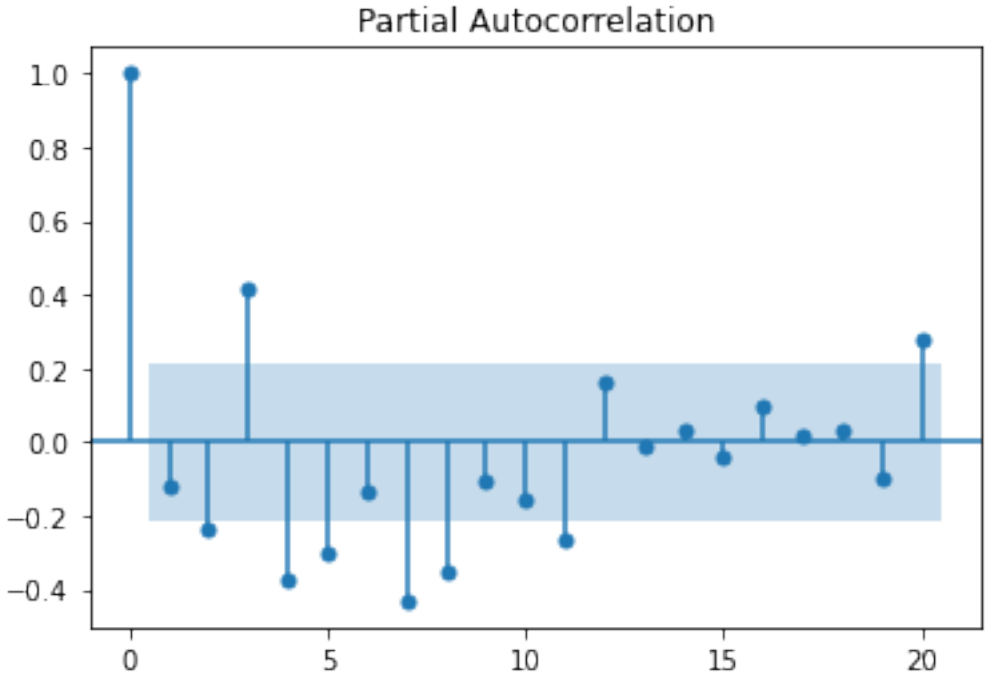
In [13]:
```
# (4)
```

In [14]:
```
# The trend line is increasing.
```

In [15]:
```
# (5)
```

In [16]:
```
first_diff = df['Miles, in Millions'].diff(1)
first_diff.dropna(inplace=True)
ACF = plot_acf(first_diff)
PACF = plot_pacf(first_diff,method="ywmle")
```

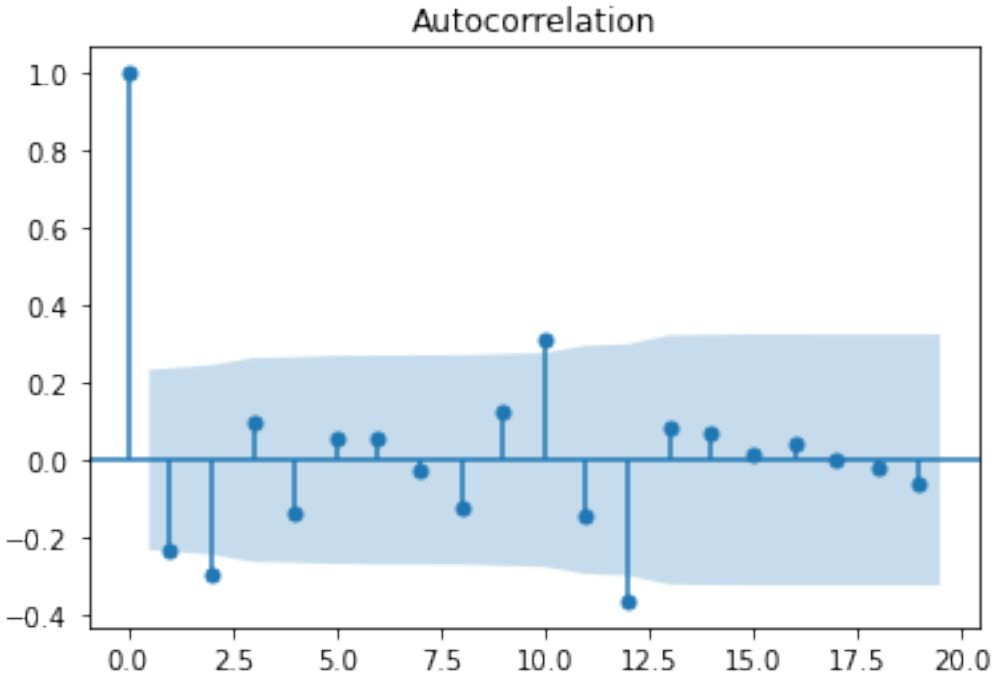


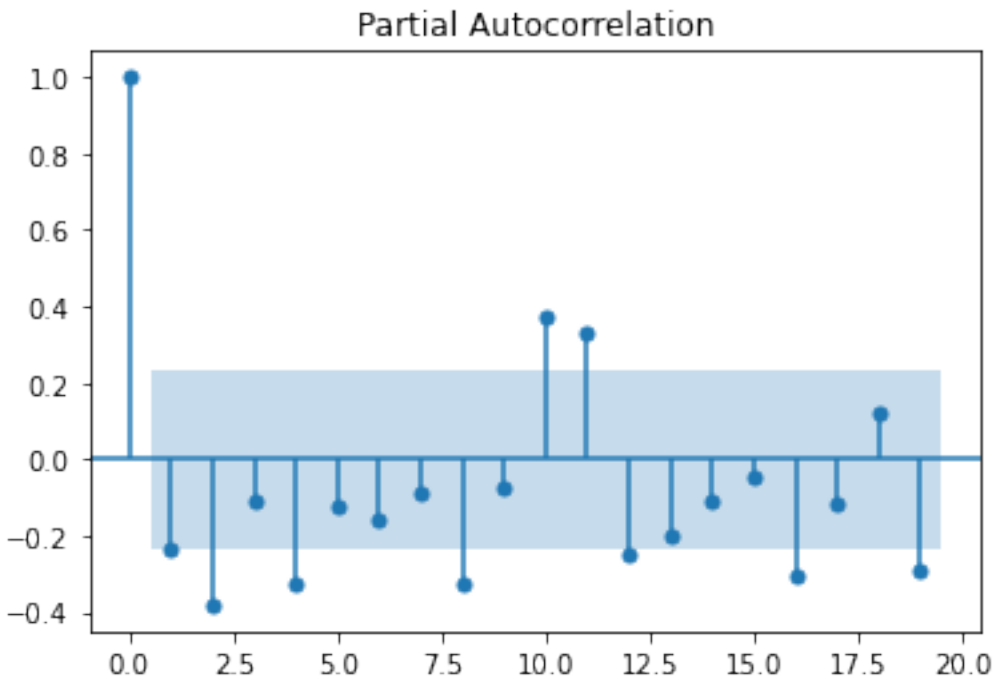


In [17]:
```
# significant lags based on ACF are 3,4,5,7,9,12.
# significant lags based on PACF is 2,3,4,5,7,8,11,20.
```

In [18]: *#(6)*

In [19]:
```
seasonal_diff = first_diff.diff(12)
seasonal_diff.dropna(inplace=True)
ACF = plot_acf(seasonal_diff)
PACF = plot_pacf(seasonal_diff)
```





In [20]:
```
# significant lags based on ACF is 2,10,12.
# significant lags based on PACF is 2,4,8,10,11,12,16,19.
```

In [21]:
```
# (7)
```

In [22]:
```
first_six_years = df[:72]
seven_years = df[72:84]
```

In [23]:
```
p = [0,1,2,3]
P = [0,1,2,3]
q = [0,1,2,3]
Q = [0,1,2,3]
d = 1
D = 1
AIC=[]
for i in p:
    for j in q:
        for m in P:
            for n in Q:
                try:
                    model = sm.tsa.statespace.SARIMAX(first_six_years['Miles, in Mill
                                                      order=(p[i],d,q[j]), seasonal_order=
                except:
                    print("p:" , p[i] , "d" , d , "q" , q[j] , "P" , P[m] , "D" , D ,
                    continue
```

```
                        model_parameter = ["p:" , p[i] , "d" , d , "q" , q[j] , "P" , P[m] , '
                        AIC.append(model_parameter)
                        AIC.append(model.aic)
            AIC_parameter = []
            for x in range(1,len(AIC),2):
                AIC_parameter.append(AIC[x])
            best_AIC = min(AIC_parameter)
```

```
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/tsa/statespace/sarimax.py:8
  warn('Too few observations to estimate starting parameters%s.'
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/tsa/statespace/sarimax.py:9
  warn('Non-stationary starting seasonal autoregressive'
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
```

```
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
```

```
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
```

```
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/tsa/statespace/sarimax.py:9
  warn('Non-invertible starting MA parameters found.'
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
```

```
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converge
  warnings.warn("Maximum Likelihood optimization failed to "
```

```
In [24]: print("The best choice of Parameters are:" , AIC[AIC.index(best_AIC)-1])
```

```
The best choice of Parameters are: ['p:', 2, 'd', 1, 'q', 3, 'P', 1, 'D', 1, 'Q', 0]
```

```
<ipython-input-24-43270c536405>:1: FutureWarning: elementwise comparison failed; returning scal
  print("The best choice of Parameters are:" , AIC[AIC.index(best_AIC)-1])
```

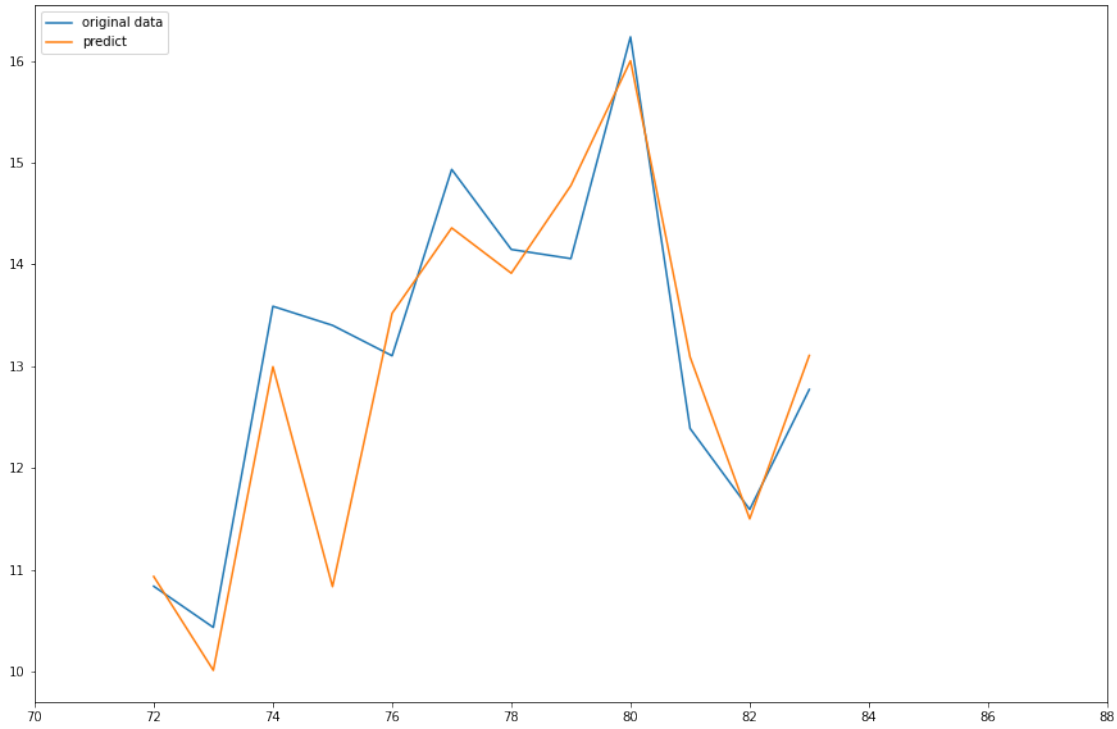
```
In [25]: best_model = sm.tsa.statespace.SARIMAX(first_six_years['Miles, in Millions'],
                                                              order=(2,1,3),seasonal_order=(1,1,0
         print(best_model.summary())
```

```
                                     SARIMAX Results
==========================================================================================
Dep. Variable:                    Miles, in Millions   No. Observations:               72
Model:             SARIMAX(2, 1, 3)x(1, 1, [], 12)   Log Likelihood              -66.642
Date:                               Tue, 26 Oct 2021   AIC                         147.283
Time:                                       02:14:50   BIC                         161.826
Sample:                                            0   HQIC                        152.960
                                                - 72
Covariance Type:                                 opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -1.4506      0.157     -9.243      0.000      -1.758      -1.143
ar.L2         -0.5747      0.149     -3.856      0.000      -0.867      -0.283
ma.L1          1.0769      5.930      0.182      0.856     -10.546      12.700
ma.L2         -0.5939      1.188     -0.500      0.617      -2.923       1.735
ma.L3         -0.8331      5.249     -0.159      0.874     -11.121       9.455
ar.S.L12      -0.4316      0.161     -2.677      0.007      -0.748      -0.116
```

```
sigma2         0.5057      3.156      0.160      0.873      -5.681       6.692
===================================================================================
Ljung-Box (L1) (Q):                   0.00   Jarque-Bera (JB):                43.23
Prob(Q):                              0.98   Prob(JB):                         0.00
Heteroskedasticity (H):               4.30   Skew:                            -0.89
Prob(H) (two-sided):                  0.00   Kurtosis:                         6.80
===================================================================================

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


/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:566: Converg
  warnings.warn("Maximum Likelihood optimization failed to "
```

```
In [26]: # I use aic as criteria.

In [27]: # (8)

In [28]: predict = best_model.forecast(12)
         predict

Out[28]: 72    10.935726
         73    10.014689
         74    12.995545
         75    10.836510
         76    13.520314
         77    14.358529
         78    13.913682
         79    14.774395
         80    15.999435
         81    13.090236
         82    11.501763
         83    13.106013
         Name: predicted_mean, dtype: float64

In [29]: SSE = (seven_years['Miles, in Millions']- predict)**2
         SSE = np.sum(SSE)
         print("SSE is ", SSE)

SSE is  8.860679881898847


In [30]: plt.figure(figsize=[15,10])
         plt.plot(seven_years['Miles, in Millions'], label='original data')
         plt.plot(predict,label='predict')
         plt.legend(loc=2)
         x_major_locator=MultipleLocator(2)
```

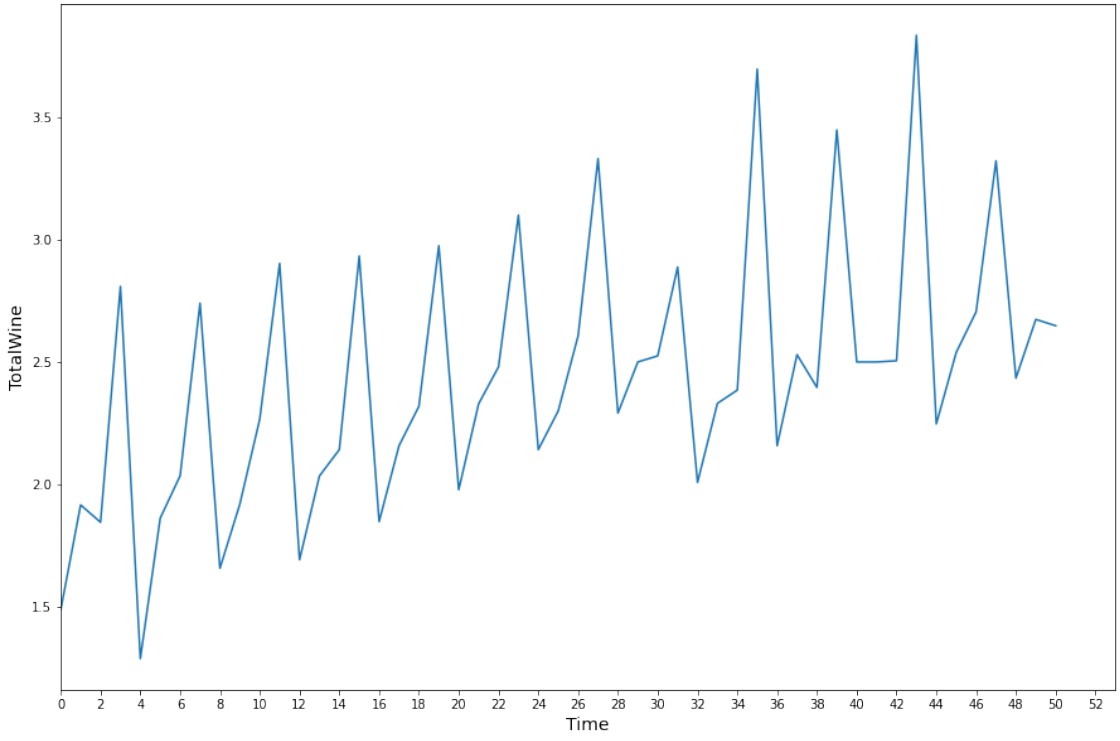
```
ax=plt.gca()
ax.xaxis.set_major_locator(x_major_locator)
plt.xlim(70,88)
plt.show()
```




```
In [31]: # From the graph, we can see that the predicted data
         # are not much different from original data.
         # Also, SSE is also great.

In [32]: # problem 2

In [33]: import numpy as np
         import pandas as pd
         import matplotlib.pyplot as plt
         from matplotlib.pyplot import MultipleLocator
         from statsmodels.graphics.tsaplots import plot_acf,plot_pacf
         import statsmodels.api as sm

In [34]: # (a)

In [35]: df = pd.read_csv("TotalWine.csv")
         df.head(10)

Out[35]:    Time (Quarter)  TotalWine
         0               1      1.486
```

```
1               2      1.915
2               3      1.844
3               4      2.808
4               5      1.287
5               6      1.861
6               7      2.034
7               8      2.739
8               9      1.656
9              10      1.918
```

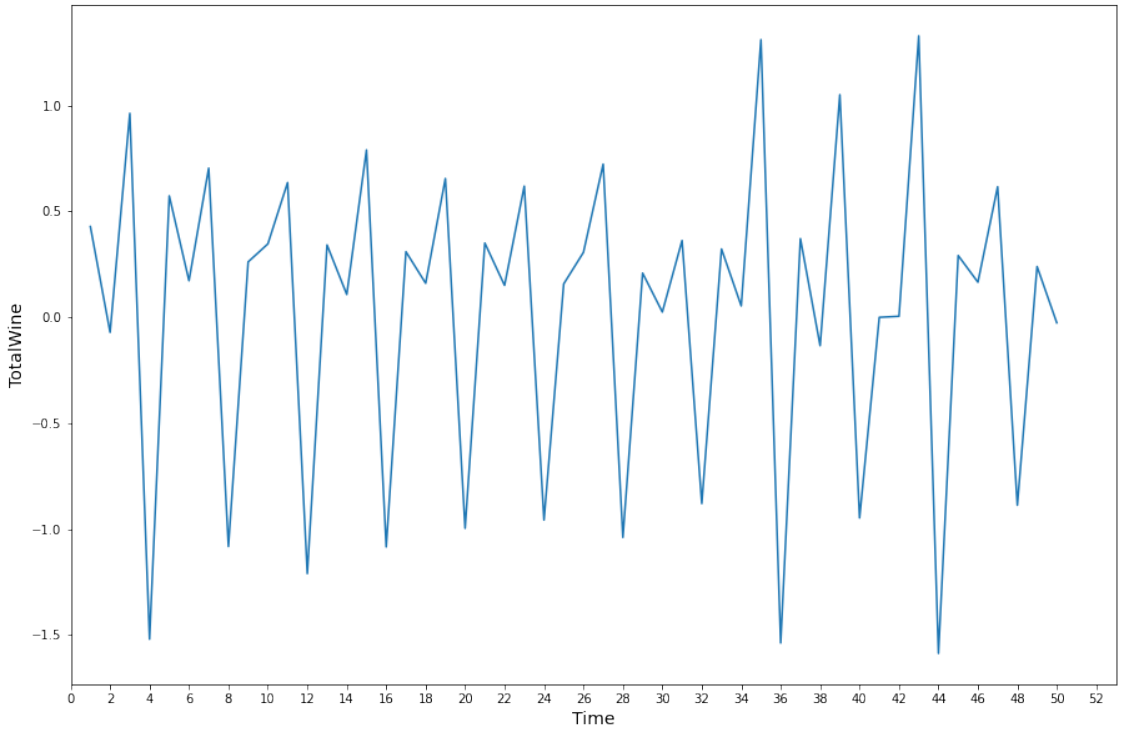
In [36]:
```
plt.figure(figsize=[15,10])
plt.plot(df['TotalWine'])
plt.xlabel('Time',fontsize=14)
plt.ylabel('TotalWine',fontsize=14)
x_major_locator=MultipleLocator(2)
ax=plt.gca()
ax.xaxis.set_major_locator(x_major_locator)
plt.xlim(0,53)
plt.show()
```



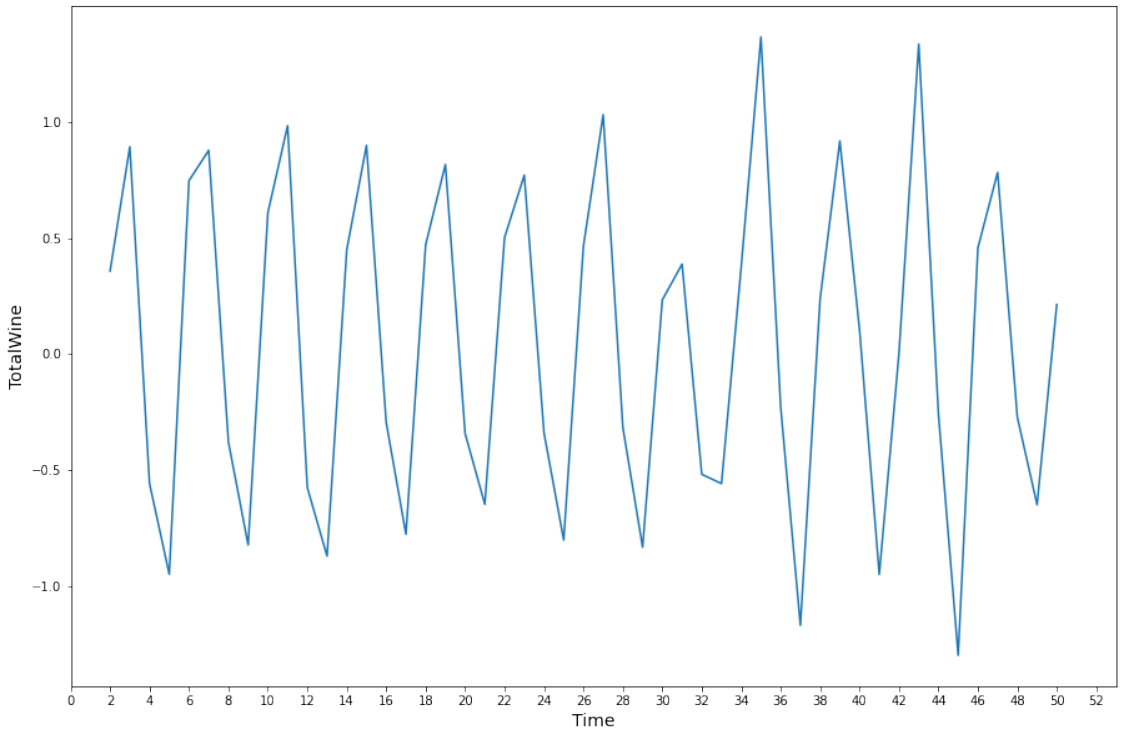
In [37]:
```
# (b)
```

In [38]:
```
first_diff = df['TotalWine'].diff(1)
first_diff.dropna(inplace=True)
```

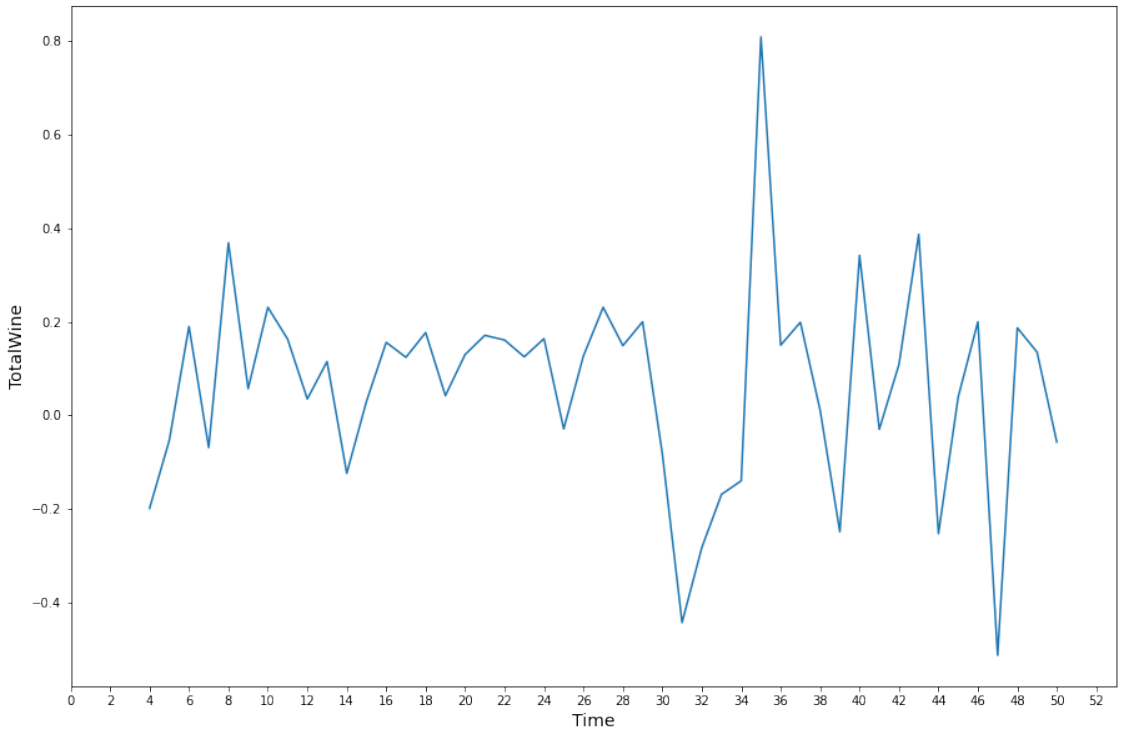
```
plt.figure(figsize=[15,10])
plt.plot(first_diff)
plt.xlabel('Time',fontsize=14)
plt.ylabel('TotalWine',fontsize=14)
x_major_locator=MultipleLocator(2)
ax=plt.gca()
ax.xaxis.set_major_locator(x_major_locator)
plt.xlim(0,53)
plt.show()
```



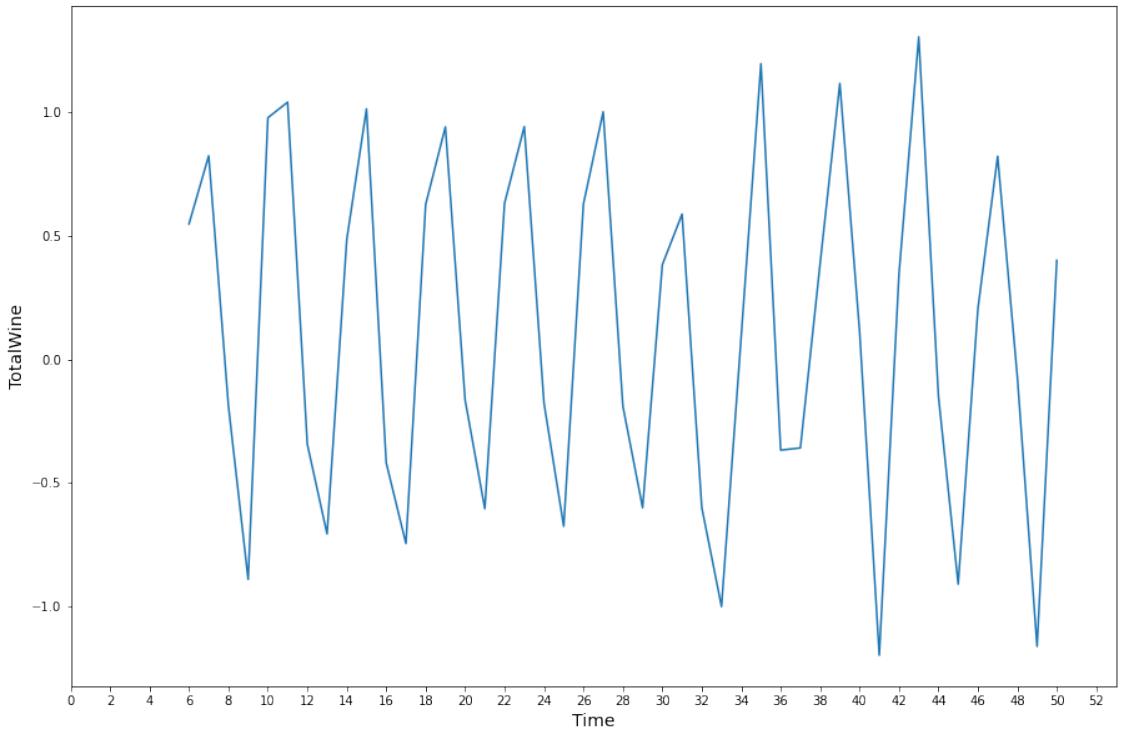
In [39]:
```
second_diff = df['TotalWine'].diff(2)
second_diff.dropna(inplace=True)
plt.figure(figsize=[15,10])
plt.plot(second_diff)
plt.xlabel('Time',fontsize=14)
plt.ylabel('TotalWine',fontsize=14)
x_major_locator=MultipleLocator(2)
ax=plt.gca()
ax.xaxis.set_major_locator(x_major_locator)
plt.xlim(0,53)
plt.show()
```



In [40]:
```
fourth_diff = df['TotalWine'].diff(4)
fourth_diff.dropna(inplace=True)
plt.figure(figsize=[15,10])
plt.plot(fourth_diff)
plt.xlabel('Time',fontsize=14)
plt.ylabel('TotalWine',fontsize=14)
x_major_locator=MultipleLocator(2)
ax=plt.gca()
ax.xaxis.set_major_locator(x_major_locator)
plt.xlim(0,53)
plt.show()
```



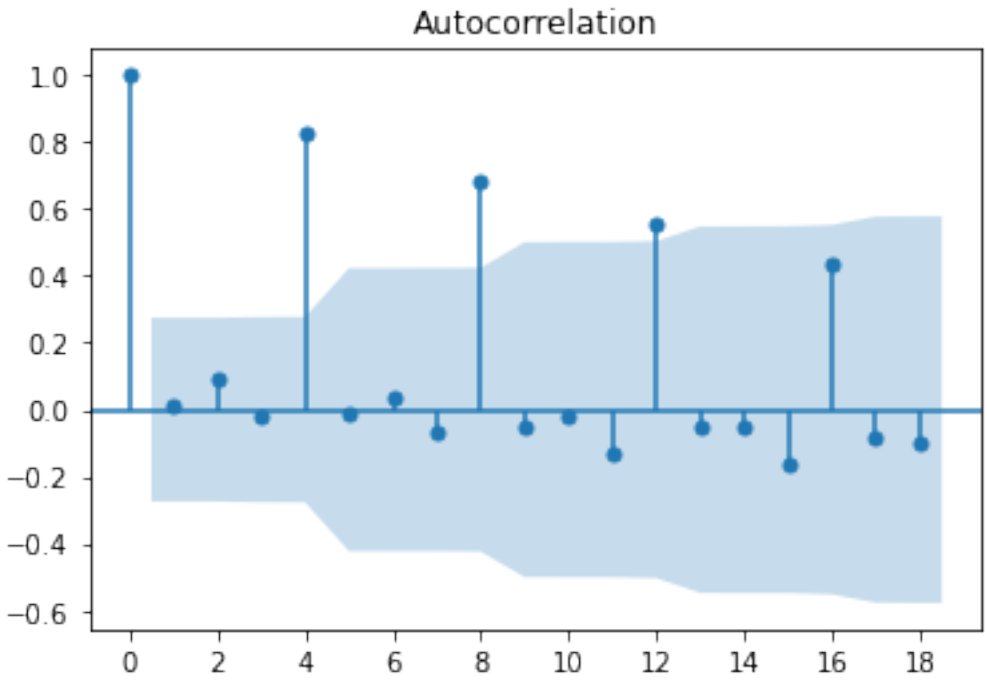
In [41]:
```
sixth_diff = df['TotalWine'].diff(6)
sixth_diff.dropna(inplace=True)
plt.figure(figsize=[15,10])
plt.plot(sixth_diff)
plt.xlabel('Time',fontsize=14)
plt.ylabel('TotalWine',fontsize=14)
x_major_locator=MultipleLocator(2)
ax=plt.gca()
ax.xaxis.set_major_locator(x_major_locator)
plt.xlim(0,53)
plt.show()
```



In [42]:
```
# lag = 4 is most suitable to remove seasonality.
```

In [43]:
```
# (c)
```

In [44]:
```
ACF = plot_acf(df['TotalWine'])
x_major_locator=MultipleLocator(2)
ax=ACF.gca()
ax.xaxis.set_major_locator(x_major_locator)
ACF.show()
```



In [45]:
```
# seasonal period is 4.
```

In [46]:
```
# (d)
```

In [47]:
```
import statsmodels.tsa.api as smt
best_order = smt.AR(df['TotalWine']).select_order(maxlag=10, ic='aic',trend='nc')
print("The best order is :", best_order)
```

```
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/tsa/ar_model.py:791: Future
statsmodels.tsa.AR has been deprecated in favor of statsmodels.tsa.AutoReg and
statsmodels.tsa.SARIMAX.

AutoReg adds the ability to specify exogenous variables, include time trends,
and add seasonal dummies. The AutoReg API differs from AR since the model is
treated as immutable, and so the entire specification including the lag
length must be specified when creating the model. This change is too
substantial to incorporate into the existing AR api. The function
ar_select_order performs lag length selection for AutoReg models.

AutoReg only estimates parameters using conditional MLE (OLS). Use SARIMAX to
estimate ARX and related models using full MLE via the Kalman Filter.

To silence this warning and continue using AR until it is removed, use:

import warnings
```

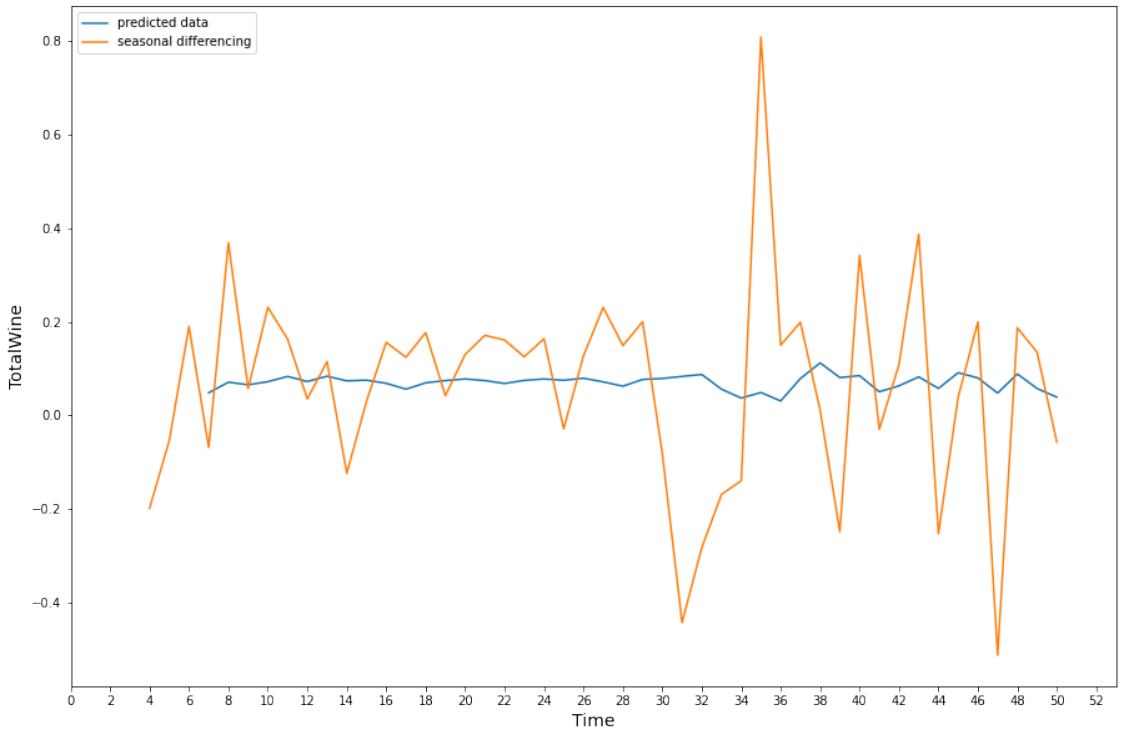
```
warnings.filterwarnings('ignore', 'statsmodels.tsa.ar_model.AR', FutureWarning)

  warnings.warn(AR_DEPRECATION_WARN, FutureWarning)
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/tsa/tsatools.py:684: Runtim
  invarcoefs = 2*np.arctanh(params)
/software/anaconda3/2020.11/lib/python3.8/site-packages/numpy/linalg/linalg.py:2099: RuntimeWa
  sign, logdet = _umath_linalg.slogdet(a, signature=signature)
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:547: Hessian
  warnings.warn('Inverting hessian failed, no bse or cov_params '
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:547: Hessian
  warnings.warn('Inverting hessian failed, no bse or cov_params '


The best order is : 3


/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:547: Hessian
  warnings.warn('Inverting hessian failed, no bse or cov_params '
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:547: Hessian
  warnings.warn('Inverting hessian failed, no bse or cov_params '
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:547: Hessian
  warnings.warn('Inverting hessian failed, no bse or cov_params '
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:547: Hessian
  warnings.warn('Inverting hessian failed, no bse or cov_params '
/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/base/model.py:547: Hessian
  warnings.warn('Inverting hessian failed, no bse or cov_params '
```

```
In [48]: # (e)
```

```
In [49]: # i
         model = smt.AR(fourth_diff).fit(3)
         print(model.summary())
```

```
                               AR Model Results
==============================================================================
Dep. Variable:                      T  -                    o                t
Model:                          AR(3)  Log Likelihood                    3.852
Method:                          cmle  S.D. of innovations               0.222
Date:                Tue, 26 Oct 2021  AIC                              -2.786
Time:                        02:14:56  BIC                              -2.583
Sample:                             0  HQIC                             -2.711

================================================================================
                  coef    std err          z      P>|z|      [0.025      0.975]
--------------------------------------------------------------------------------
const           0.0664      0.040      1.660      0.097      -0.012       0.145
L1.TotalWine   -0.0269      0.158     -0.170      0.865      -0.336       0.283
L2.TotalWine    0.0302      0.157      0.192      0.848      -0.278       0.339
```

```
L3.TotalWine     0.0575       0.155       0.370       0.711       -0.247       0.362
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1            2.4829           -0.0000j            2.4829           -0.0000
AR.2           -1.5037           -2.1769j            2.6458           -0.3462
AR.3           -1.5037           +2.1769j            2.6458            0.3462
-----------------------------------------------------------------------------


/software/anaconda3/2020.11/lib/python3.8/site-packages/statsmodels/tsa/base/tsa_model.py:578:
  warnings.warn('An unsupported index was provided and will be'


In [50]: # ii
         predict=model.predict()
         predict.head(10)

Out[50]: 7     0.048183
         8     0.070846
         9     0.065304
         10    0.071999
         11    0.083111
         12    0.072237
         13    0.083635
         14    0.073711
         15    0.075181
         16    0.068435
         dtype: float64

In [51]: # iii
         plt.figure(figsize=[15,10])
         plt.plot(predict,label='predicted data')
         plt.xlabel('Time',fontsize=14)
         plt.ylabel('TotalWine',fontsize=14)
         x_major_locator=MultipleLocator(2)
         ax=plt.gca()
         ax.xaxis.set_major_locator(x_major_locator)
         plt.xlim(0,53)
         plt.plot(fourth_diff,label='seasonal differencing')
         plt.legend(loc=2)
         plt.show()
```




```
In [52]: # iv
         MAE = (abs(fourth_diff-predict)).sum()
         print(MAE/51)
```

```
0.13847755793145622
```