Project1_TimeSeries

October 29, 2019

Importing all the necessary libraries

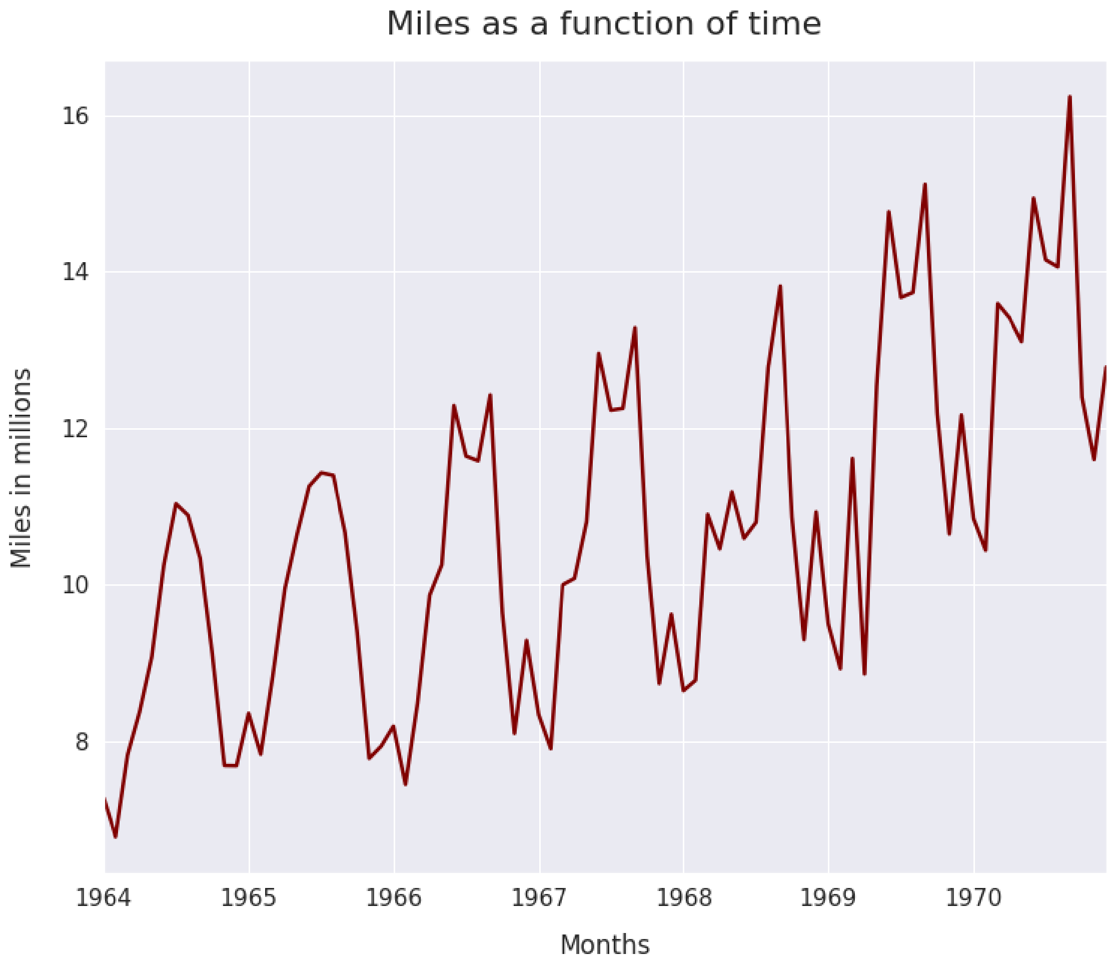
```
In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        from datetime import datetime
        import seaborn as sns
        from statsmodels.tsa.arima_model import ARIMA
        from statsmodels.tsa.statespace.sarimax import SARIMAX
        from statsmodels.graphics.tsaplots import plot_acf
        from statsmodels.graphics.tsaplots import plot_pacf
        sns.set(font_scale=1.4)
        import math
```

Importing the data in and converting the date coulmn into a date time series

```
In [2]: data = pd.read_excel("Project1_DataSet.xlsx")
        data['Month'] = pd.to_datetime(data['Month'])
        series = data["Miles, in Millions"]
        time = data['Month']
```

1 Q1 Solution

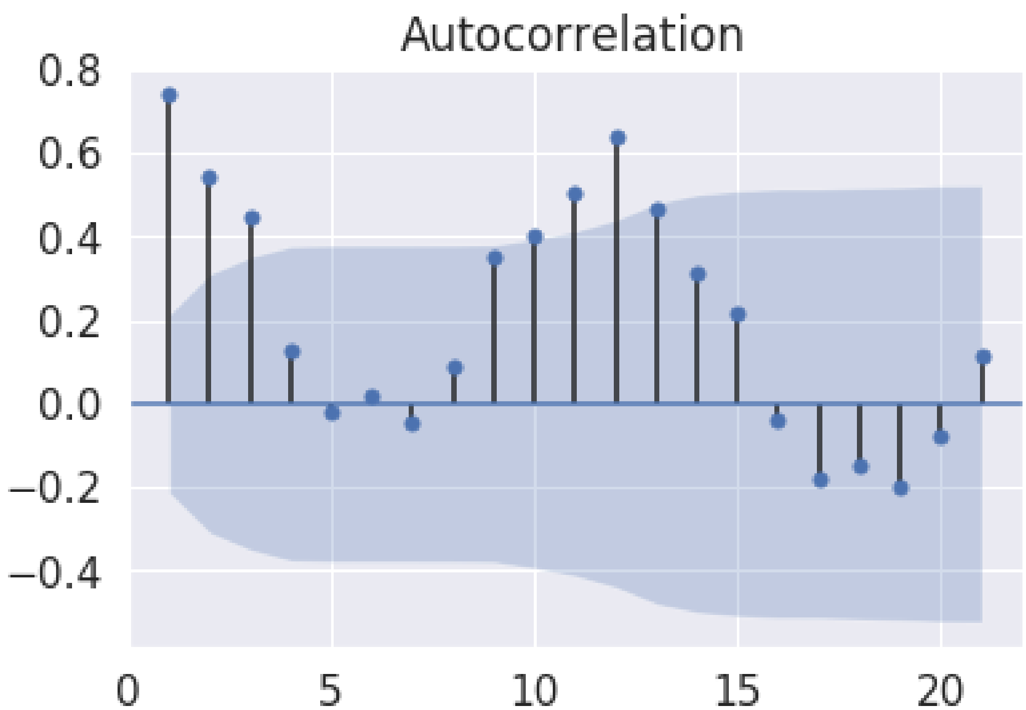
```
In [3]: data.set_index('Month')['Miles, in Millions'].plot(figsize=(12, 10), linewidth=2.5, co
        plt.xlabel("Months", labelpad=15)
        plt.ylabel("Miles in millions", labelpad=15)
        plt.title("Miles as a function of time", y=1.02, fontsize=22)

Out[3]: Text(0.5, 1.02, 'Miles as a function of time')
```



2 Q2 Solution

2.0.1 ACF plotted for the series.

```
In [4]: n_lags = np.round(len(series)/4)
        plot_acf(series,lags=n_lags, zero=False)
        plt.show()
```



2.0.2 Q2

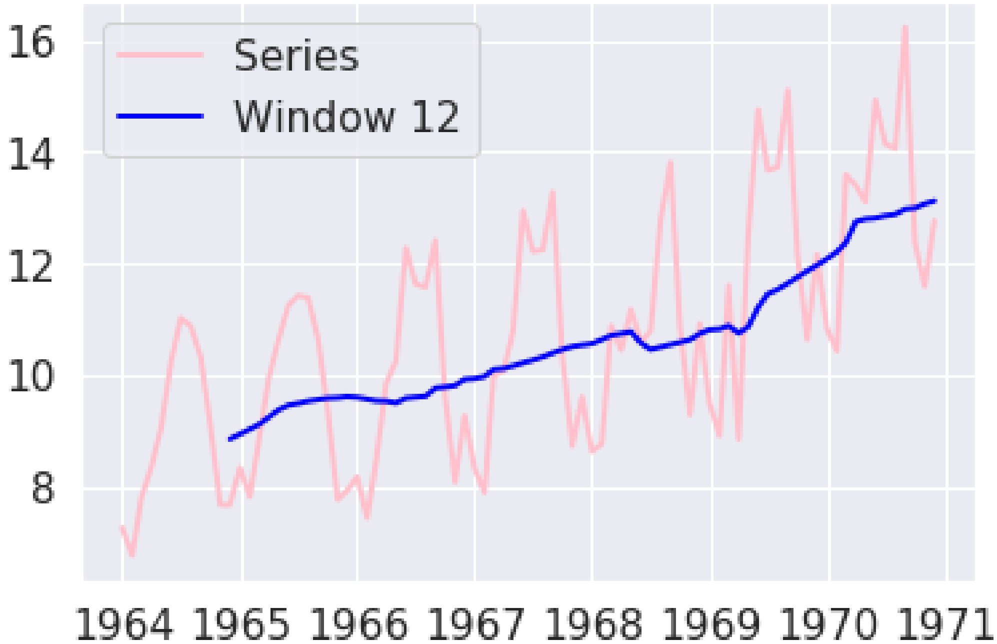
2.0.3 The season period is 12 months.

2.0.4 The highest significant lag in the ACF is at 12 which denotes the seasonal period (except lag 1).

3 Q3 Solution

In [5]:
```
ma_13 = (series).rolling(window=12).mean()
plt.plot( time, series, color='pink', linewidth=2, label= "Series")
plt.plot( time, ma_13,marker='', color='blue', linewidth=2, label="Window 12")
plt.legend()
```

Out[5]:
```
<matplotlib.legend.Legend at 0x2ba569db71d0>
```



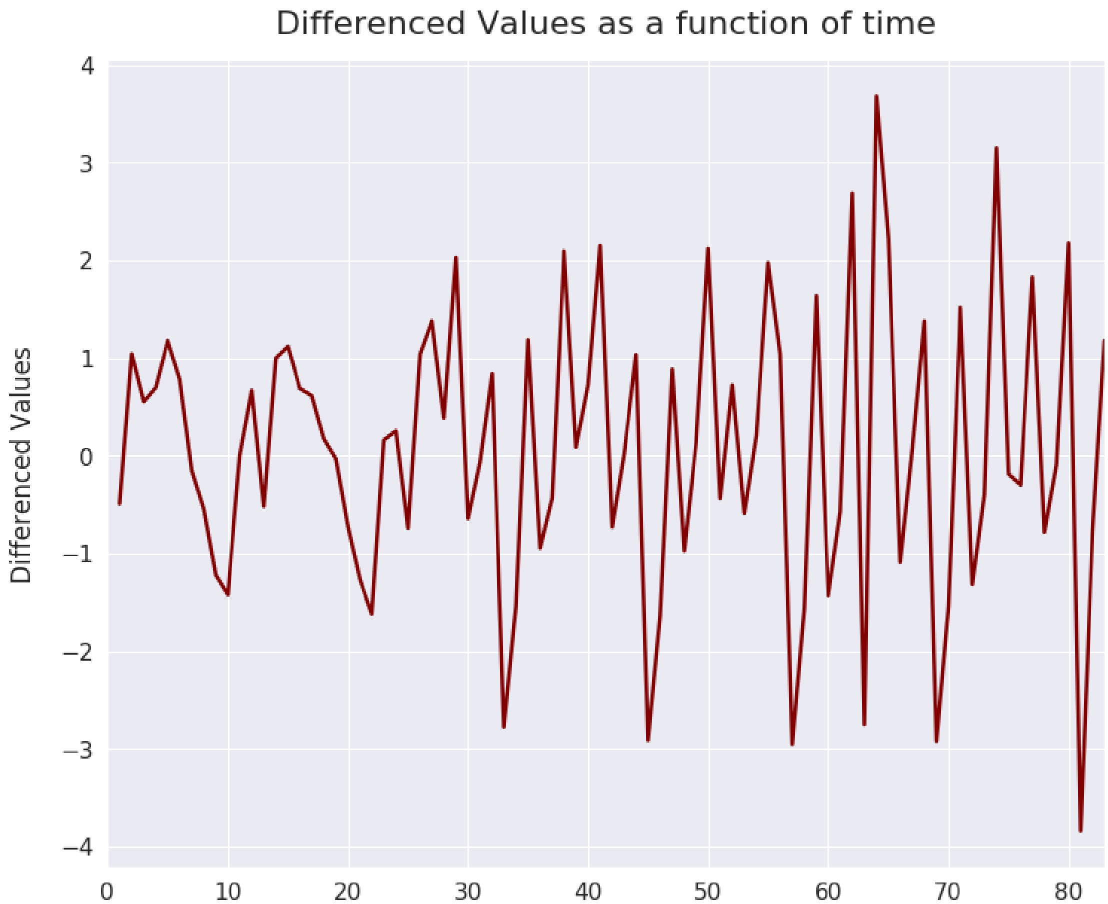
3.0.1 The moving average window should be 12 since the seasonal period is 12 months. It also provdies a smoother plot than a 13 month window.

4 Q4 Solution

4.0.1 There is an increasing trending as we can clearly see from the moving average and time series graph.

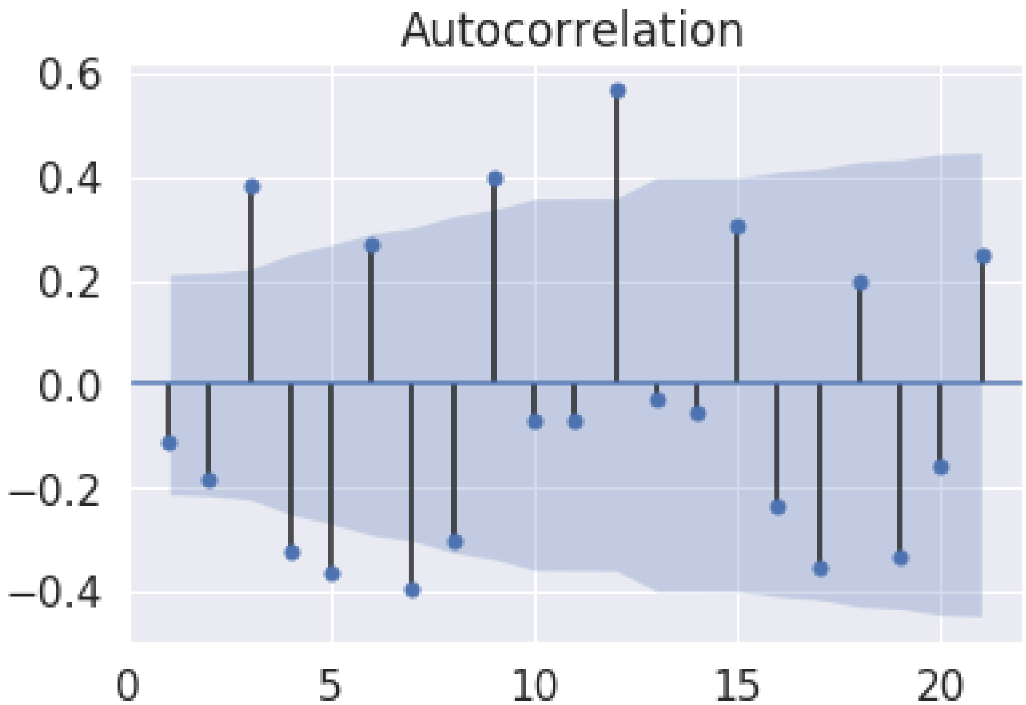
5 Q5 Solution

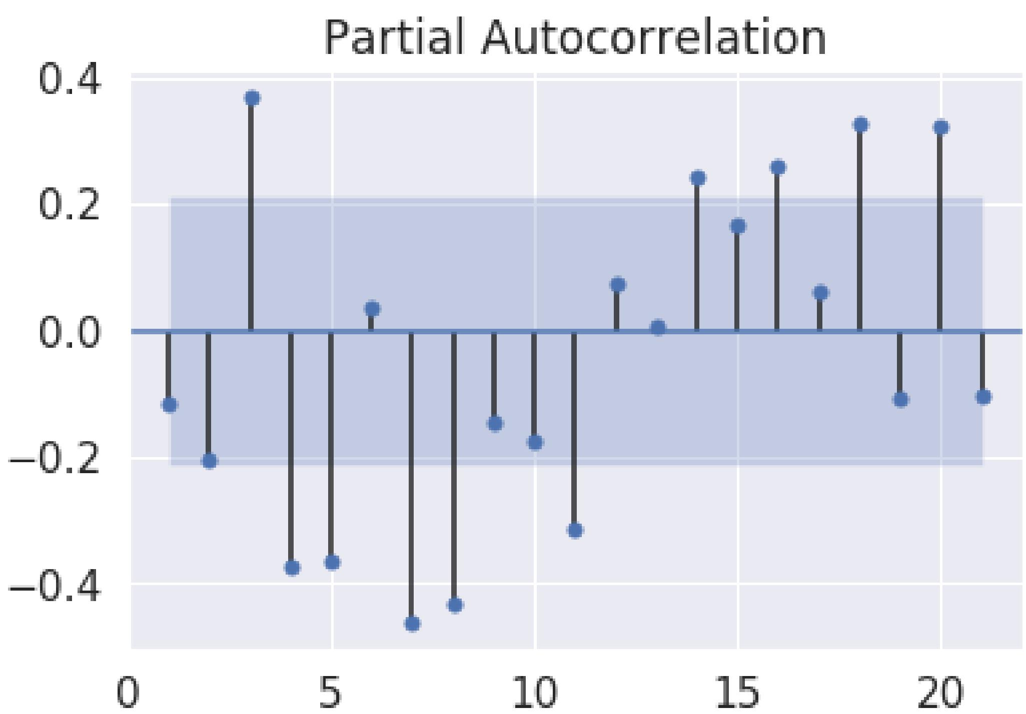
```
In [6]: diff = series.diff()
        diff.plot(figsize=(12, 10), linewidth=2.5, color='maroon')
        plt.ylabel("Differenced Values", labelpad=15)
        plt.title("Differenced Values as a function of time", y=1.02, fontsize=22)
        plt.show()
```



5.0.1 Plotting the ACF and the PACF of the differences series

```
In [7]: diff[[0]] = 0
        n_lags = np.round(len(diff)/4)
        plot_acf(diff.astype(int),lags=n_lags,zero=False)
        plot_pacf(diff.astype(int),lags=n_lags,zero=False)
        plt.show()
```


Autocorrelation
0.6
0.4
0.2
0.0
−0.2
−0.4
0
5
10
15
20

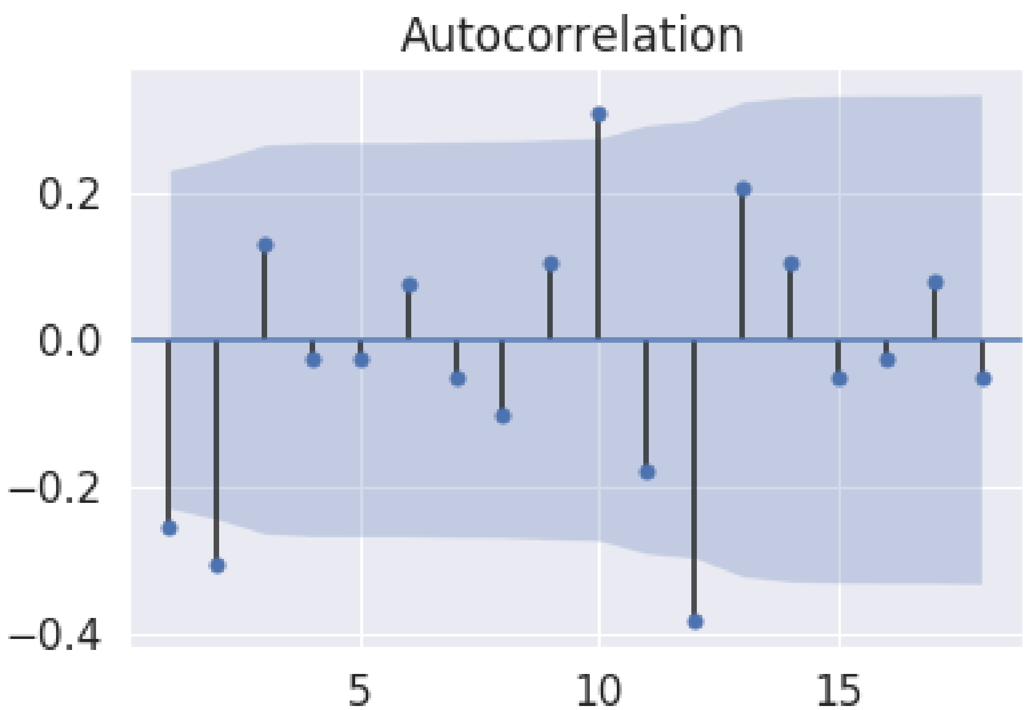

Partial Autocorrelation
0.4
0.2
0.0
−0.2
−0.4
0
5
10
15
20

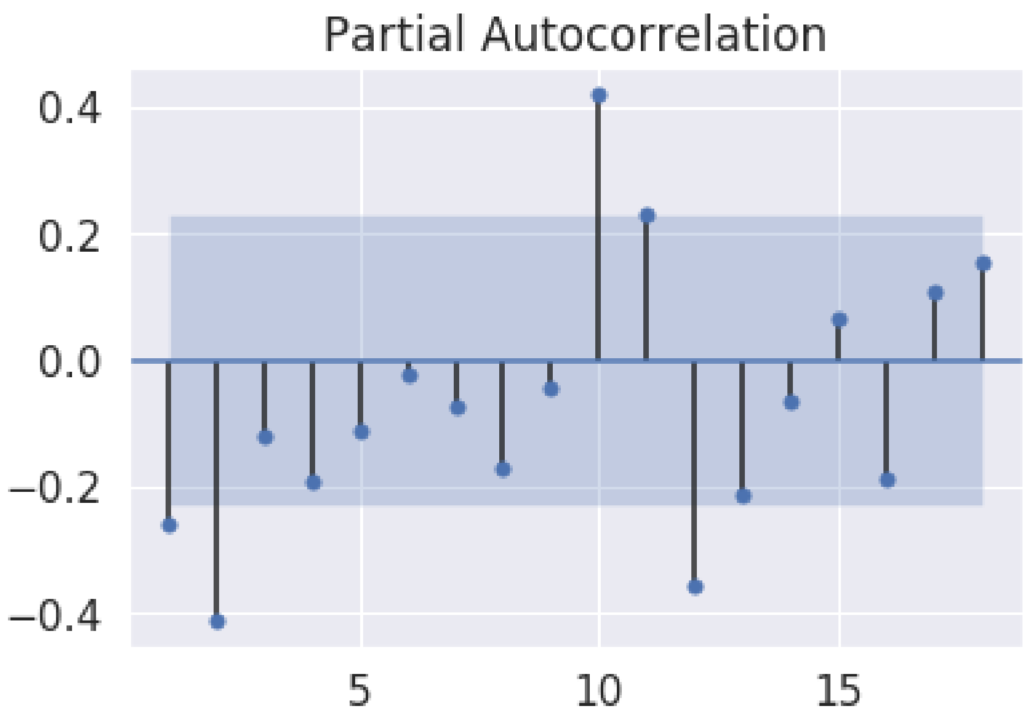
5.0.2 The significant lags in the ACF are 3, 4, 5, 7, 9 and 12

5.0.3 The significant lags in the PACF are 3, 4, 5, 7, 8, 11 and more

6 Q6 Solution

In [8]:
```
seasonal_diff = diff -diff.shift(12)
n_lags = np.round(len(seasonal_diff[12:])/4)
plot_acf(seasonal_diff[12:].astype(int),lags=n_lags,zero=False)
plot_pacf(seasonal_diff[12:].astype(int),lags=n_lags,zero=False)
plt.show()
```





6.0.1 The significant lags in the ACF are 1, 2, 10 and 12

6.0.2 The significant lags in the PACF are 1, 2, 10 and 12

7 Q7

7.0.1 From question 4 will know that we need to apply first order differencing since there is a trend and from question 6 we know that the AR model order needs to be between 1 and 4 while the MA model order also needs to be between 1 and 4. Along with the seasonal difference.

```
In [9]: score = math.inf

        Order = []
        Seasonal_Order = []
        AIC_score = []

        for p in range(0,5):
            for q in range(0,5):
                for P in range(0,5):
                    for Q in range(0,5):

                        try:
```

```
                    o = (p,1,q)
                    s_order = (P,1,Q,12)

                    model = SARIMAX(series[:72], order = o, seasonal_order=s_order)
                    results = model.fit()
                    AIC = results.aic
                    AIC_score.append(AIC)

                    Order.append(o)

                    Seasonal_Order.append(s_order)

                    if AIC < score:
                        score = AIC
                        #print(AIC)
                        best_order = o
                        best_seasonal_order = s_order
                    else:
                        pass
                except:
                    pass



print("The best order is",best_order)

print("The best seasonal order is",best_seasonal_order)

print("The lowest AIC is",score)
```

```
/software/anaconda3/2019.03/lib/python3.7/site-packages/statsmodels/tsa/statespace/representati
  return matrix[[slice(None)]*(matrix.ndim-1) + [0]]
/software/anaconda3/2019.03/lib/python3.7/site-packages/statsmodels/base/model.py:508: Converge
  "Check mle_retvals", ConvergenceWarning)
/software/anaconda3/2019.03/lib/python3.7/site-packages/statsmodels/base/model.py:508: Converge
  "Check mle_retvals", ConvergenceWarning)
/software/anaconda3/2019.03/lib/python3.7/site-packages/statsmodels/base/model.py:508: Converge
  "Check mle_retvals", ConvergenceWarning)
/software/anaconda3/2019.03/lib/python3.7/site-packages/statsmodels/base/model.py:508: Converge
  "Check mle_retvals", ConvergenceWarning)
/software/anaconda3/2019.03/lib/python3.7/site-packages/statsmodels/base/model.py:508: Converge
  "Check mle_retvals", ConvergenceWarning)
/software/anaconda3/2019.03/lib/python3.7/site-packages/statsmodels/base/model.py:508: Converge
  "Check mle_retvals", ConvergenceWarning)
/software/anaconda3/2019.03/lib/python3.7/site-packages/statsmodels/base/model.py:508: Converge
  "Check mle_retvals", ConvergenceWarning)
/software/anaconda3/2019.03/lib/python3.7/site-packages/statsmodels/base/model.py:508: Converge
  "Check mle_retvals", ConvergenceWarning)
```

```
/software/anaconda3/2019.03/lib/python3.7/site-packages/statsmodels/base/model.py:508: Converge
  "Check mle_retvals", ConvergenceWarning)
/software/anaconda3/2019.03/lib/python3.7/site-packages/statsmodels/base/model.py:508: Converge
  "Check mle_retvals", ConvergenceWarning)
/software/anaconda3/2019.03/lib/python3.7/site-packages/statsmodels/base/model.py:508: Converge
  "Check mle_retvals", ConvergenceWarning)
/software/anaconda3/2019.03/lib/python3.7/site-packages/statsmodels/base/model.py:508: Converge
  "Check mle_retvals", ConvergenceWarning)
/software/anaconda3/2019.03/lib/python3.7/site-packages/statsmodels/base/model.py:508: Converge
  "Check mle_retvals", ConvergenceWarning)
/software/anaconda3/2019.03/lib/python3.7/site-packages/statsmodels/base/model.py:508: Converge
  "Check mle_retvals", ConvergenceWarning)
/software/anaconda3/2019.03/lib/python3.7/site-packages/statsmodels/base/model.py:508: Converge
  "Check mle_retvals", ConvergenceWarning)
/software/anaconda3/2019.03/lib/python3.7/site-packages/statsmodels/base/model.py:508: Converge
  "Check mle_retvals", ConvergenceWarning)
/software/anaconda3/2019.03/lib/python3.7/site-packages/statsmodels/base/model.py:508: Converge
  "Check mle_retvals", ConvergenceWarning)
/software/anaconda3/2019.03/lib/python3.7/site-packages/statsmodels/base/model.py:508: Converge
  "Check mle_retvals", ConvergenceWarning)
/software/anaconda3/2019.03/lib/python3.7/site-packages/statsmodels/base/model.py:508: Converge
  "Check mle_retvals", ConvergenceWarning)
/software/anaconda3/2019.03/lib/python3.7/site-packages/statsmodels/base/model.py:508: Converge
  "Check mle_retvals", ConvergenceWarning)
/software/anaconda3/2019.03/lib/python3.7/site-packages/statsmodels/base/model.py:508: Converge
  "Check mle_retvals", ConvergenceWarning)
/software/anaconda3/2019.03/lib/python3.7/site-packages/statsmodels/base/model.py:508: Converge
  "Check mle_retvals", ConvergenceWarning)
/software/anaconda3/2019.03/lib/python3.7/site-packages/statsmodels/base/model.py:508: Converge
  "Check mle_retvals", ConvergenceWarning)
/software/anaconda3/2019.03/lib/python3.7/site-packages/statsmodels/base/model.py:508: Converge
  "Check mle_retvals", ConvergenceWarning)
/software/anaconda3/2019.03/lib/python3.7/site-packages/statsmodels/base/model.py:508: Converge
  "Check mle_retvals", ConvergenceWarning)
/software/anaconda3/2019.03/lib/python3.7/site-packages/statsmodels/base/model.py:508: Converge
  "Check mle_retvals", ConvergenceWarning)
/software/anaconda3/2019.03/lib/python3.7/site-packages/statsmodels/base/model.py:508: Converge
  "Check mle_retvals", ConvergenceWarning)
/software/anaconda3/2019.03/lib/python3.7/site-packages/statsmodels/base/model.py:508: Converge
  "Check mle_retvals", ConvergenceWarning)
/software/anaconda3/2019.03/lib/python3.7/site-packages/statsmodels/base/model.py:508: Converge
  "Check mle_retvals", ConvergenceWarning)
/software/anaconda3/2019.03/lib/python3.7/site-packages/statsmodels/base/model.py:508: Converge
  "Check mle_retvals", ConvergenceWarning)
```

```
/software/anaconda3/2019.03/lib/python3.7/site-packages/statsmodels/base/model.py:508: Converge
  "Check mle_retvals", ConvergenceWarning)
/software/anaconda3/2019.03/lib/python3.7/site-packages/statsmodels/base/model.py:508: Converge
  "Check mle_retvals", ConvergenceWarning)
/software/anaconda3/2019.03/lib/python3.7/site-packages/statsmodels/base/model.py:508: Converge
  "Check mle_retvals", ConvergenceWarning)
/software/anaconda3/2019.03/lib/python3.7/site-packages/statsmodels/base/model.py:508: Converge
  "Check mle_retvals", ConvergenceWarning)
/software/anaconda3/2019.03/lib/python3.7/site-packages/statsmodels/base/model.py:508: Converge
  "Check mle_retvals", ConvergenceWarning)
/software/anaconda3/2019.03/lib/python3.7/site-packages/statsmodels/base/model.py:508: Converge
  "Check mle_retvals", ConvergenceWarning)
/software/anaconda3/2019.03/lib/python3.7/site-packages/statsmodels/base/model.py:508: Converge
  "Check mle_retvals", ConvergenceWarning)
/software/anaconda3/2019.03/lib/python3.7/site-packages/statsmodels/base/model.py:508: Converge
  "Check mle_retvals", ConvergenceWarning)
/software/anaconda3/2019.03/lib/python3.7/site-packages/statsmodels/base/model.py:508: Converge
  "Check mle_retvals", ConvergenceWarning)


The best order is (2, 1, 3)
The best seasonal order is (1, 1, 0, 12)
The lowest AIC is 147.28340552146926
```

In [10]:
```
AIC_table  = pd.DataFrame()
AIC_table['Order'] = Order
AIC_table['Seasonal Order'] = Seasonal_Order
AIC_table['AIC'] = AIC_score
AIC_table.sort_values(by=['AIC'])
```

Out[10]:

	Order	Seasonal Order	AIC
93	(2, 1, 3)	(1, 1, 0, 12)	147.283406
92	(2, 1, 3)	(0, 1, 1, 12)	147.796169
16	(0, 1, 2)	(1, 1, 0, 12)	148.968651
99	(2, 1, 4)	(0, 1, 1, 12)	148.986842
15	(0, 1, 2)	(0, 1, 1, 12)	149.224977
95	(2, 1, 3)	(2, 1, 0, 12)	149.236361
100	(2, 1, 4)	(1, 1, 0, 12)	149.457987
94	(2, 1, 3)	(1, 1, 1, 12)	149.578717
121	(3, 1, 3)	(1, 1, 0, 12)	149.699652
51	(1, 1, 2)	(1, 1, 0, 12)	149.858222
50	(1, 1, 2)	(0, 1, 1, 12)	149.995964
23	(0, 1, 3)	(1, 1, 0, 12)	150.102492
14	(0, 1, 2)	(0, 1, 0, 12)	150.105497
22	(0, 1, 3)	(0, 1, 1, 12)	150.170875
120	(3, 1, 3)	(0, 1, 1, 12)	150.440980
18	(0, 1, 2)	(2, 1, 0, 12)	150.964349

```
44   (1, 1, 1)  (1, 1, 0, 12)   151.040104
17   (0, 1, 2)  (1, 1, 1, 12)   151.043874
96   (2, 1, 3)  (3, 1, 0, 12)   151.140487
102  (2, 1, 4)  (2, 1, 0, 12)   151.185886
30   (0, 1, 4)  (1, 1, 0, 12)   151.310332
29   (0, 1, 4)  (0, 1, 1, 12)   151.549688
147  (4, 1, 3)  (0, 1, 1, 12)   151.647972
123  (3, 1, 3)  (2, 1, 0, 12)   151.684866
43   (1, 1, 1)  (0, 1, 1, 12)   151.698195
91   (2, 1, 3)  (0, 1, 0, 12)   151.723697
133  (4, 1, 1)  (0, 1, 1, 12)   151.768707
53   (1, 1, 2)  (2, 1, 0, 12)   151.771099
86   (2, 1, 2)  (1, 1, 0, 12)   151.857864
58   (1, 1, 3)  (1, 1, 0, 12)   151.858090
..         ...            ...          ...
83   (2, 1, 1)  (3, 1, 1, 12)   158.005940
152  (4, 1, 3)  (3, 1, 1, 12)   158.066265
131  (4, 1, 0)  (3, 1, 1, 12)   158.113051
109  (3, 1, 0)  (2, 1, 0, 12)   158.127160
75   (2, 1, 0)  (3, 1, 0, 12)   158.128318
108  (3, 1, 0)  (1, 1, 1, 12)   158.145728
118  (3, 1, 1)  (3, 1, 1, 12)   158.230148
66   (1, 1, 4)  (1, 1, 1, 12)   158.689664
69   (1, 1, 4)  (3, 1, 1, 12)   159.073136
105  (3, 1, 0)  (0, 1, 0, 12)   159.554038
145  (4, 1, 2)  (3, 1, 1, 12)   159.698557
76   (2, 1, 0)  (3, 1, 1, 12)   160.073622
110  (3, 1, 0)  (3, 1, 0, 12)   160.127077
7    (0, 1, 1)  (0, 1, 0, 12)   160.295406
111  (3, 1, 0)  (3, 1, 1, 12)   162.070138
37   (1, 1, 0)  (1, 1, 0, 12)   163.307685
36   (1, 1, 0)  (0, 1, 1, 12)   163.789152
39   (1, 1, 0)  (2, 1, 0, 12)   165.132139
38   (1, 1, 0)  (1, 1, 1, 12)   165.210748
2    (0, 1, 0)  (1, 1, 0, 12)   165.333070
1    (0, 1, 0)  (0, 1, 1, 12)   165.608301
40   (1, 1, 0)  (3, 1, 0, 12)   166.409535
0    (0, 1, 0)  (0, 1, 0, 12)   166.902178
35   (1, 1, 0)  (0, 1, 0, 12)   167.195685
4    (0, 1, 0)  (2, 1, 0, 12)   167.321479
3    (0, 1, 0)  (1, 1, 1, 12)   167.327523
41   (1, 1, 0)  (3, 1, 1, 12)   168.408456
5    (0, 1, 0)  (3, 1, 0, 12)   168.571740
6    (0, 1, 0)  (3, 1, 1, 12)   170.533949
24   (0, 1, 3)  (1, 1, 1, 12)  1223.309950

[153 rows x 3 columns]
```

8 Q8 Solution

```
In [11]: model = SARIMAX(series[:72], order = best_order, seasonal_order=best_seasonal_order)
         model_fit = model.fit()
         #print(model_fit.summary())
         yhat = model_fit.forecast(12)
         error = np.sum(np.square(yhat-series[-12:]))/(12)
         print("The mean squared error for forecast is",error)
```

```
The mean squared error for forecast is 0.7383706098643396
```

```
In [12]: print("Dataframe with actual and forecasted: \n")
         forecast = pd.DataFrame()
         forecast['Ground Truth'] = series[-12:]
         forecast['Forecasted'] = yhat
         print(forecast)
```

```
Dataframe with actual and forecasted:

    Ground Truth  Forecasted
72        10.840   10.936182
73        10.436   10.014992
74        13.589   12.995767
75        13.402   10.836447
76        13.103   13.520746
77        14.933   14.359624
78        14.147   13.914507
79        14.057   14.774762
80        16.234   15.999929
81        12.389   13.090693
82        11.594   11.502247
83        12.772   13.106475
```

8.0.1 The forecast can be improved if we had more data by using another model order. We are not able to explore many combinations of model orders because of the scarcity of data.