EE 627 - Data Acquisition & Processing I

HW - 2

Date : $13^{th}$ Fab 2021

Name :- Hinalben Patel
CWID :- 10473912

Problem :- 1 Justify the below statements are True or false.

(a) For a stationary AR(1) time series $x(t)$, $x(t)$ is uncorrelated to $x(t-l)$ for $l \geq 2$.

Ans :- $AR(1) = x(t) = a_0 + a_1 x_{t-1} + \varepsilon_t = a(aX_{t-2} + \varepsilon_{t-1}) + \varepsilon_t$

Lag 2 : $x(t-2) = a x_{t-3} + \varepsilon_{t-2}$.

Now, $x(t) = a^3 x(t-3) + a^2 \varepsilon_{t-2} + a \varepsilon_{t-1} + \varepsilon_t$

$x(t-2) = a\, x(t-3) + \varepsilon_{t-2}$

we can seen that the terms are common in both eq^n

So, $x(t)$ and $x(t-2)$ are correlated

So, the statement is false.

(b) For a Stationary MA(1) time series $x(t)$, you will observe a coefficient cliff after time lag $l >= 1$ in the ACF plot

Ans :- $x(t) = \theta \varepsilon_{t-1} + \varepsilon_t + \mu$.

$x(t-1) = \theta \varepsilon_{t-2} + \varepsilon_{t-1} + \mu$

$x(t-2) = \theta \varepsilon_{t-3} + \varepsilon_{t-2} + \mu$

→ from that above equations, there is no common terms in all three $x(t)$, $x(t-1)$ & $x(t-2)$ for MA(1)

→ So, the equations are uncorrelated, ACF is also 0 for $l > 1$

→ The statement is False

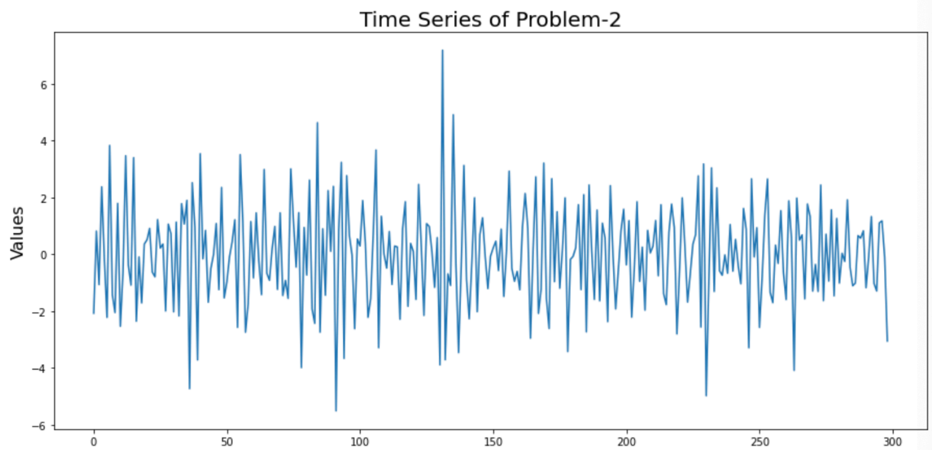
Scanned with CamScanner

**Find the best predictive model for below time series using the techniques in the lecture.**

**Problem 2:**

- Plot the time series for Data given in CSV File

```
In [98]: plt.figure(figsize=(15,7))

         plt.plot(data)

         plt.title('Time Series of Problem-2', fontsize=20)
         plt.ylabel('Values', fontsize=16)

Out[98]: Text(0, 0.5, 'Values')
```

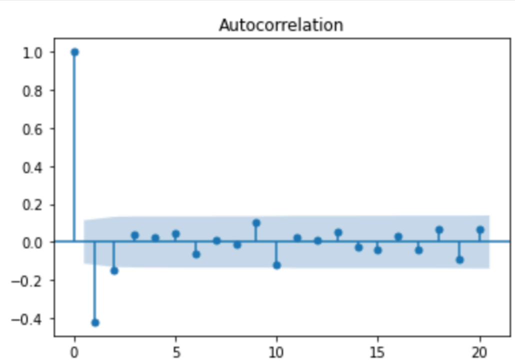



Above graph shows that the time series is stationary.

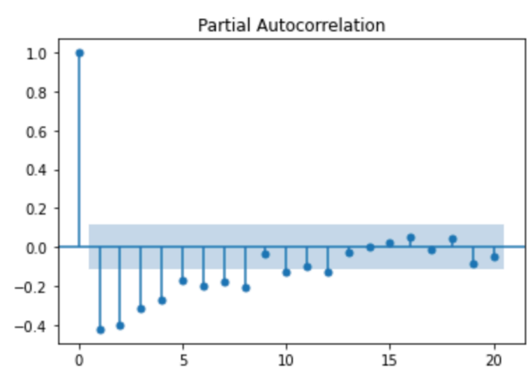
- ACF and PACF

ACF

```
In [26]: acf_plot = plot_acf(data, lags=20)
```




```
In [100]: pacf_plot = plot_pacf(data, lags=20)
```

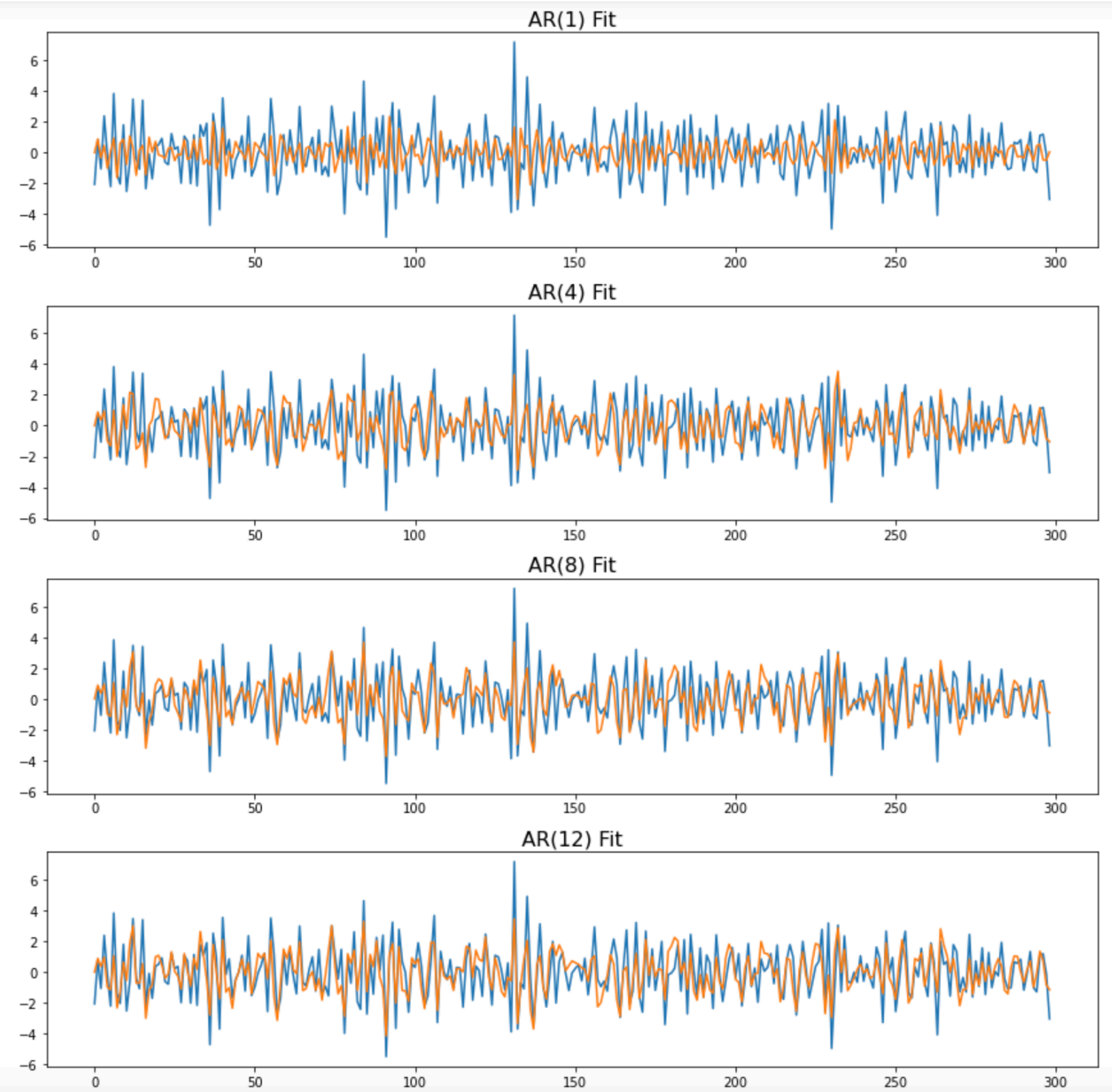



For an AR model, in ACF, there are two strong lags and others are too small. So we can see that in the plot ACF to a declination behavior over time. For MA models, we observe the ACF plot.

For an MA model, It is opposite from AR model. We would expect the PACF to a diminishing behavior over time, which can seen in plot. For AR models, we observe the PACF plot.

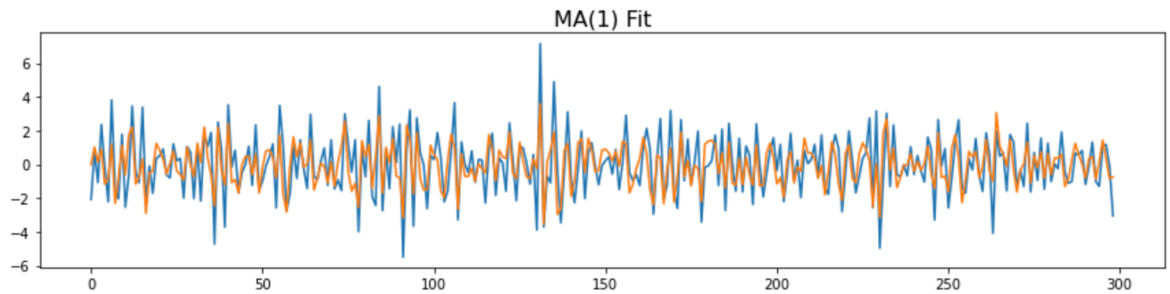
Based on the above graph, we have start with a moving Average Model with lags 1 and 2. Also, we can try Autoregressive Model with lags 1,4,8 and 12.

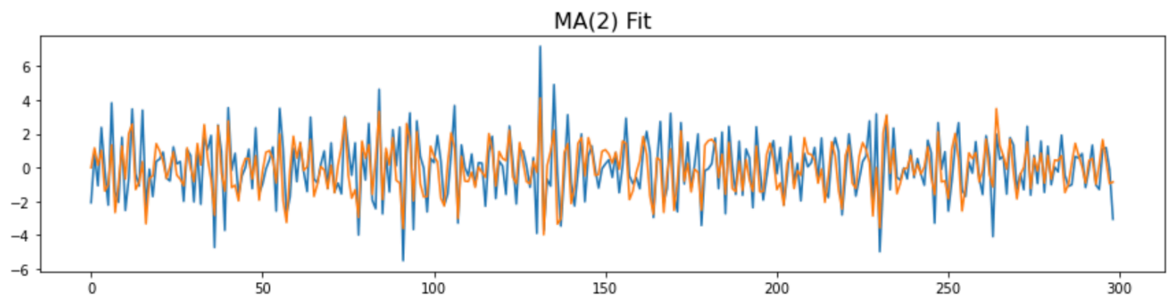
- AR(p) Model


AR(1) Fit
AR(4) Fit
AR(8) Fit
AR(12) Fit
6
4
2
0
−2
−4
−6
0
50
100
150
200
250
300

- MA(q) Model

In [31]:
```
plt.figure(figsize=(12, 12))

ma_orders = [1, 2]
ma_fitted_model_dict = {}

for idx, ma_order in enumerate(ma_orders):
    # Create MA(q) model
    ma_model = ARMA(data, order=(0, ma_order))
    ma_model_fit = ma_model.fit()
    ma_fitted_model_dict[ma_order] = ma_model_fit
    plt.subplot(4, 1, idx + 1)
    plt.plot(data)
    plt.plot(ma_model_fit.fittedvalues)    # fitted values are predicted values from the model
    plt.title('MA(%s) Fit'%ma_order, fontsize=16)

plt.tight_layout()
```

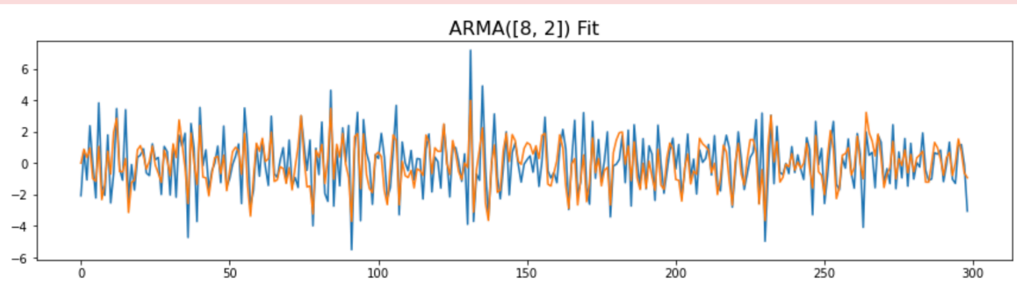



- Finally, try an ARMA Model with the best lags from each other(8 from AR and 2 from MA)

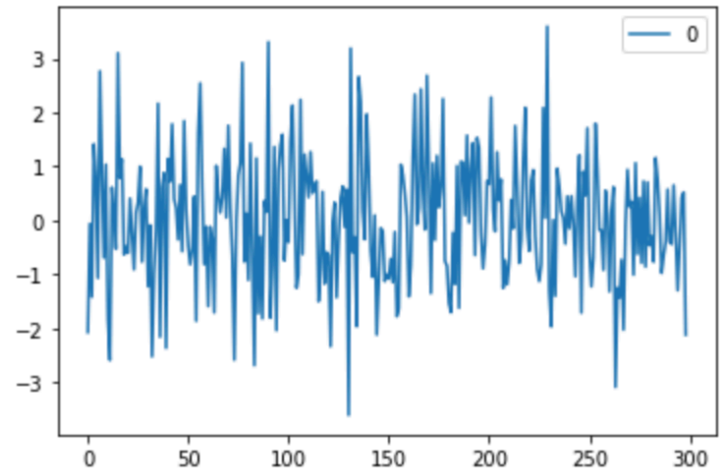
In [32]:
```
plt.figure(figsize=(12, 12))
arma_order = [8, 2]

# Create ARMA(p,q) model
arma_model = ARMA(data, order=(arma_order[0], arma_order[1]))
arma_model_fit = arma_model.fit()
plt.subplot(4, 1, 1)
plt.plot(data)
plt.plot(arma_model_fit.fittedvalues)    # fitted values are predicted values from the model
plt.title('ARMA(%s) Fit'%arma_order, fontsize=16)

plt.tight_layout()
```

```
/Users/hinalpatel/opt/anaconda3/lib/python3.8/site-packages/statsmodels/base/model.py:547: He
ssianInversionWarning: Inverting hessian failed, no bse or cov_params available
  warnings.warn('Inverting hessian failed, no bse or cov_params '
```

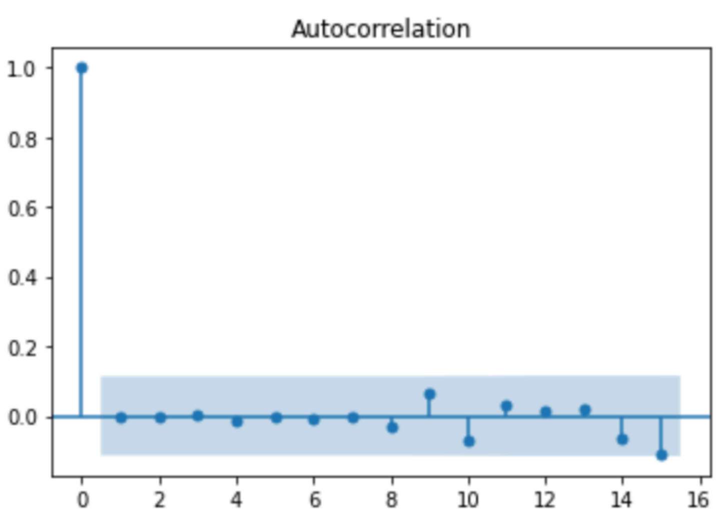


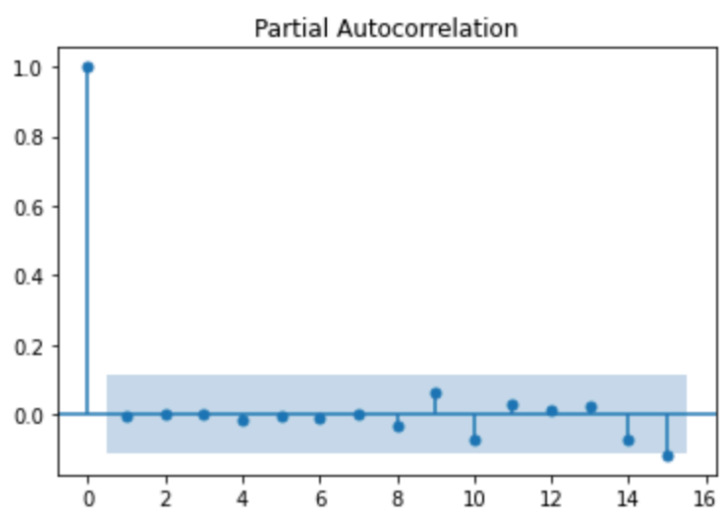
- For checking out, The model is good fit or not, we should plot residuals(difference between raw time series data and fitted model)

```
In [101]: arma = pd.DataFrame(arma_model_fit.resid)
          arma.plot()
          plt.show()
```




- After that we should plot ACF and PACF, so that we can see if they are equal to zero for all lags.




From above plot we can see that all values are around zero for ACF and PACF, so we can say that it is a very good model.

- AIC comparison

```
In [149]: for ar_order in ar_orders:
              print('AIC for AR(%s): %s'%(ar_order, fitted_model_dict[ar_order].aic))

AIC for AR(1): 1155.8657264616293
AIC for AR(4): 1052.892247082635
AIC for AR(8): 1014.3860891695313
AIC for AR(12): 1004.2241095579899
```

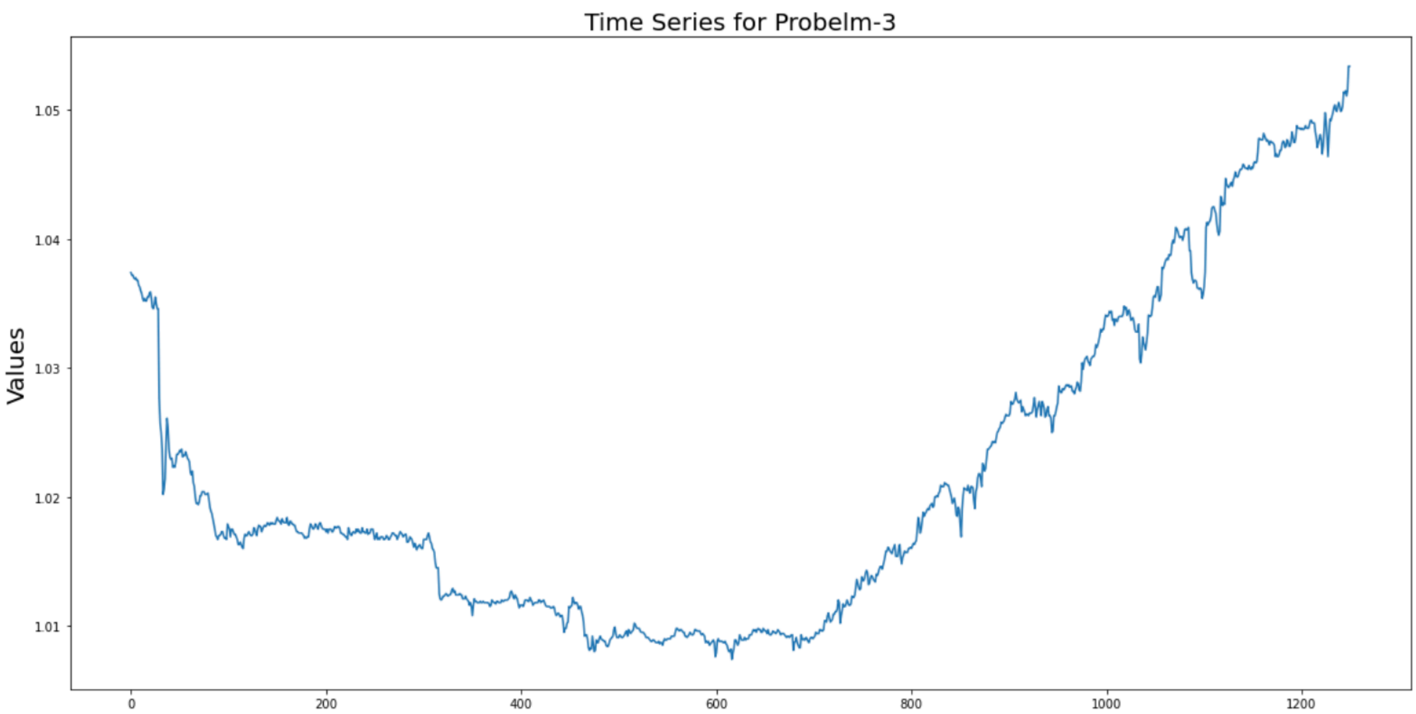
```
In [150]: for ma_order in ma_orders:
              print('AIC for MA(%s): %s'%(ma_order, ma_fitted_model_dict[ma_order].aic))

AIC for MA(1): 994.078121654427
AIC for MA(2): 993.2166661414136
```

```
In [151]: print('AIC for ARMA(%s): %s'%(arma_order, arma_model_fit.aic))

AIC for ARMA([8, 2]): 986.1461905561986
```

From this AIC values, We can say that an AR(12) model, MA(2) model or ARMA(8,2) model will be best, because it's value of AIC are smaller than other model's AIC values.

**Problem 3:**

- Plot the time series for Data given in CSV File

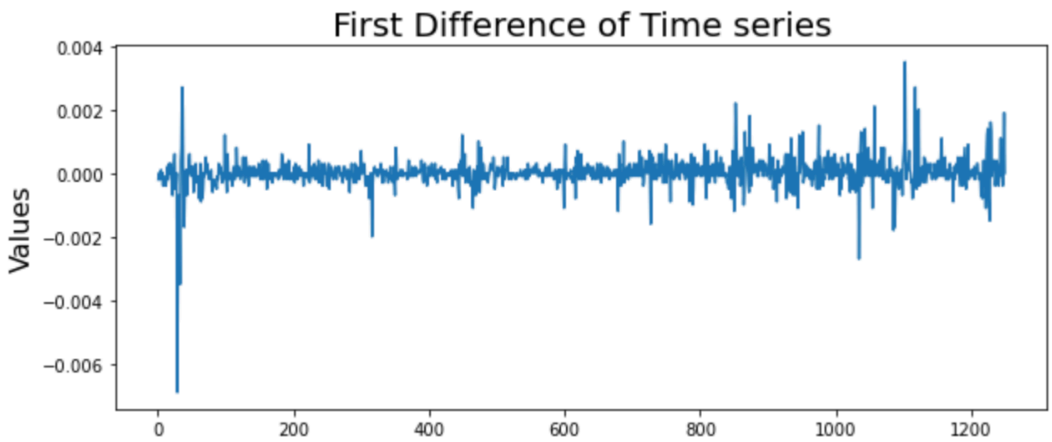



Above graph shows that, time series is not stationary. So, we have try to transform it something that is stationary

- Firstly, take the First difference, then plot it.

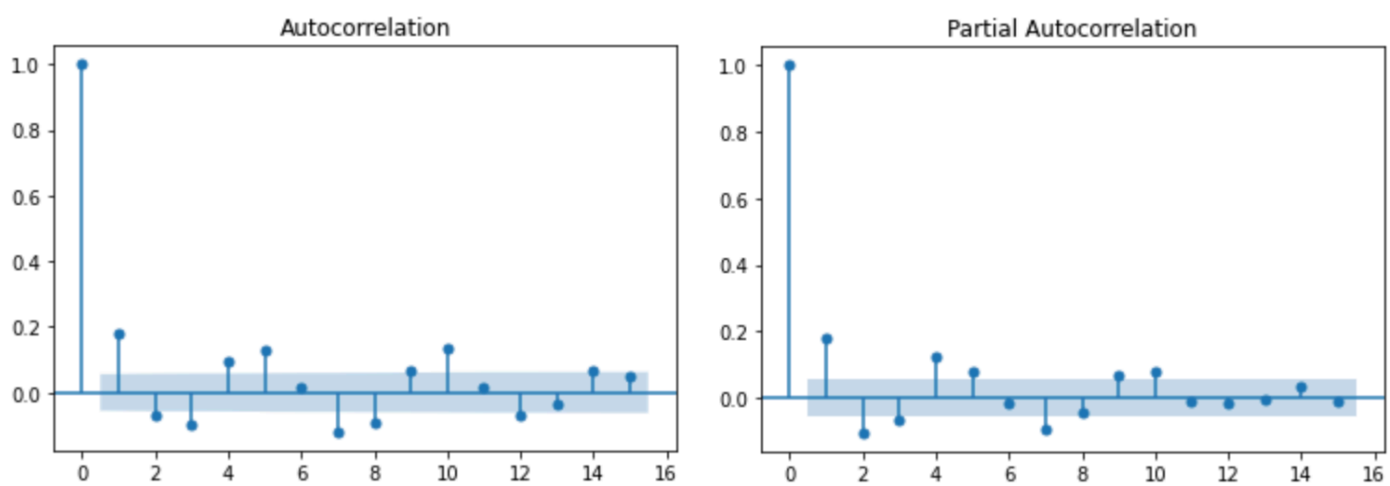
In [113]:
```
first_diff = df.diff()[1:]
plt.figure(figsize=(10, 4))
plt.plot(first_diff)
plt.title('First Difference of Time series', fontsize=20)
plt.ylabel('Values', fontsize=16)
```

Out[113]: Text(0, 0.5, 'Values')



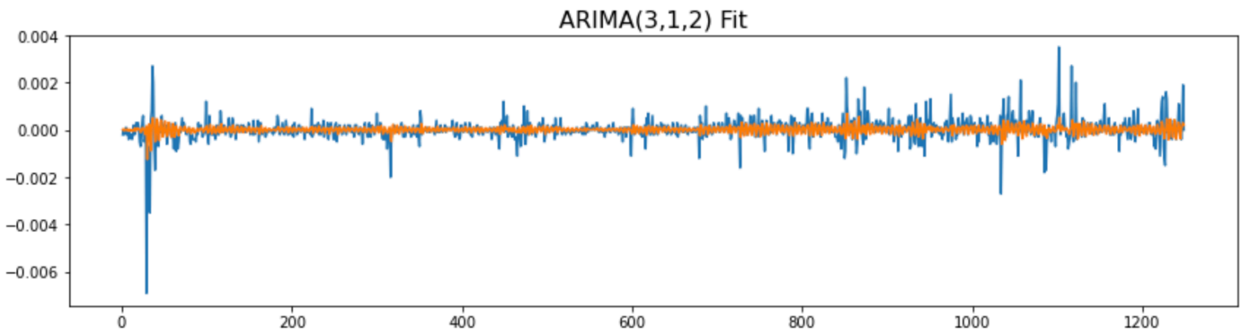

Now, The graph looks as Stationary.

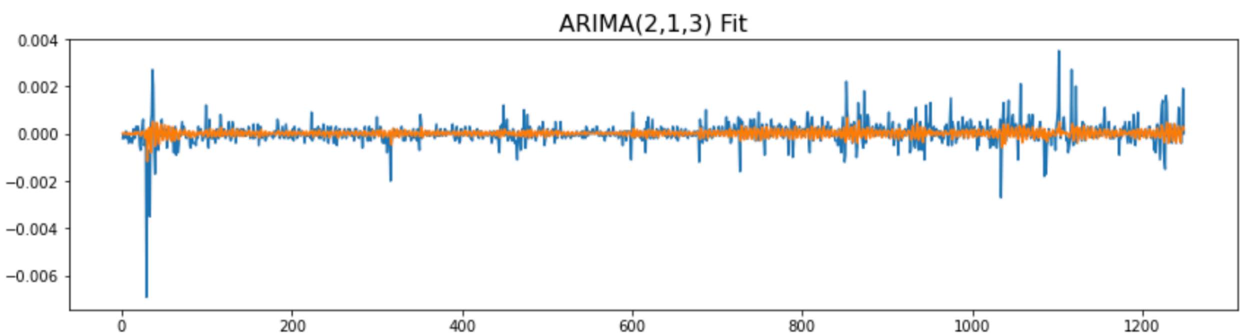
- Now, lets plot ACF and PACF for time series



- Plot ARIMA(p,d,q) model

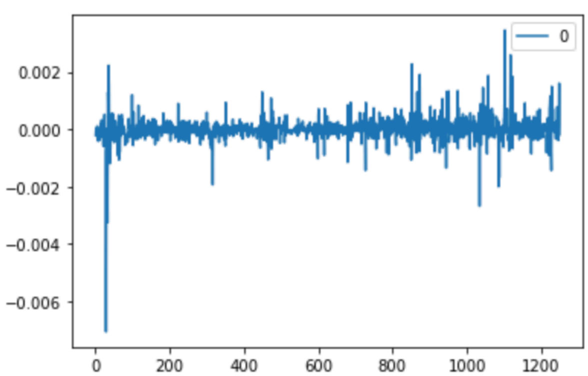
In [133]:
```
plt.figure(figsize=(12,12))
order_1 = (3, 1, 2)

# Create ARIMA(p,d,q) model
arima_model_1 = ARIMA(df[0], order=(order_1[0], order_1[1], order_1[2]))
arima_model_fit_1 = arima_model_1.fit()
plt.subplot(4, 1, 1)
plt.plot(first_diff)
plt.plot(arima_model_fit_1.fittedvalues)
plt.title('ARIMA(3,1,2) Fit', fontsize=16)

plt.tight_layout()

plt.figure(figsize=(12, 12))
order_2 = (2, 1, 3)

# Create ARIMA(p,d,q) model
arima_model_2 = ARIMA(df[0], order=(order_2[0], order_2[1], order_2[2]))
arima_model_fit_2 = arima_model_2.fit()
plt.subplot(4, 1, 1)
plt.plot(first_diff)
plt.plot(arima_model_fit_2.fittedvalues)
plt.title('ARIMA(2,1,3) Fit', fontsize=16)

plt.tight_layout()
```

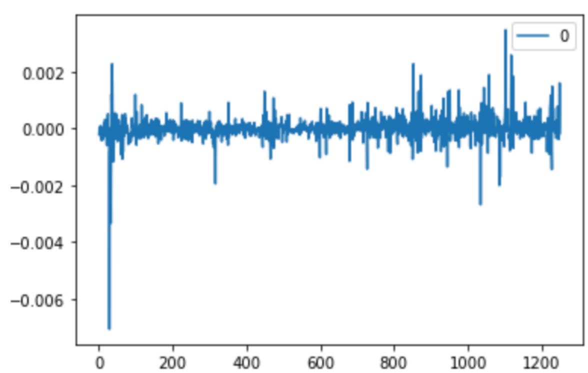



- Plot the residuals for both model

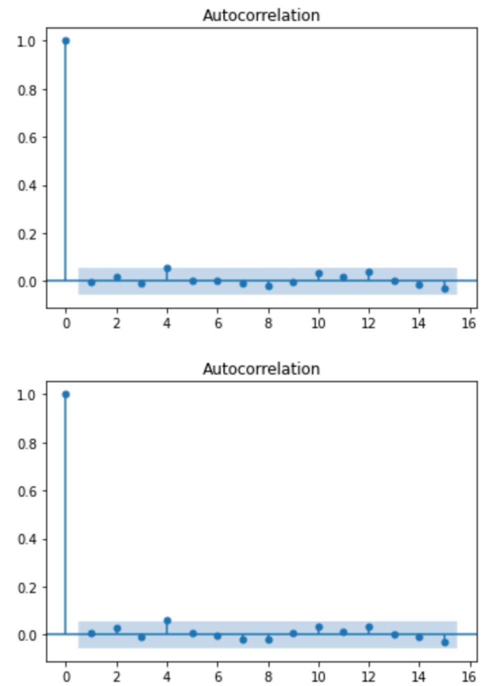
In [91]:
```
arima_residuals_1 = pd.DataFrame(arima_model_fit_1.resid)
arima_residuals_1.plot()
arima_residuals_2 = pd.DataFrame(arima_model_fit_2.resid)
arima_residuals_2.plot()
plt.show()
```



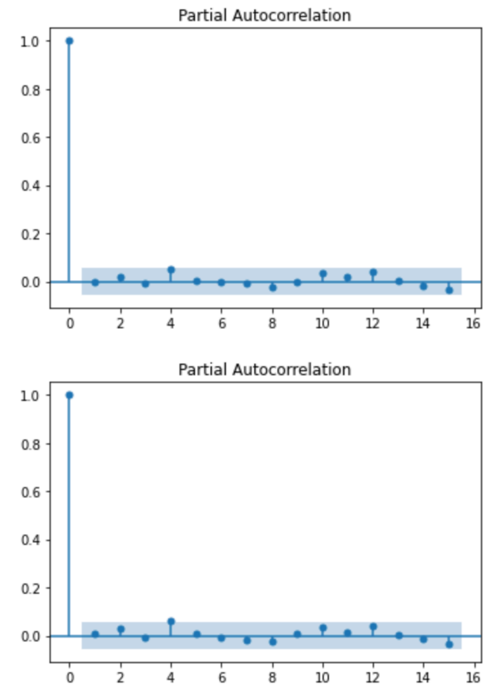



- Residual ACF and PACF for both ARIMA Model

In [92]:
```
residual_acf_plot_1 = plot_acf(arima_residuals_1, lags=15)
residual_acf_plot_2 = plot_acf(arima_residuals_2, lags=15)
```




In [93]:
```
residual_pacf_plot_1 = plot_pacf(arima_residuals_1, lags=15)
residual_pacf_plot_2 = plot_pacf(arima_residuals_2, lags=15)
```




- AIC for ARIMA:

In [94]:
```
print('AIC for ARIMA(3,1,2): %s'%(arima_model_fit_1.aic))
print('AIC for ARIMA(2,1,3): %s'%(arima_model_fit_2.aic))
```

```
AIC for ARIMA(3,1,2): -15676.53208916025
AIC for ARIMA(2,1,3): -15674.93408606686
```

AIC values are very close from each other, but AIC for ARIMA(3,1,2) is slightly smaller, so that is the slightly better model.