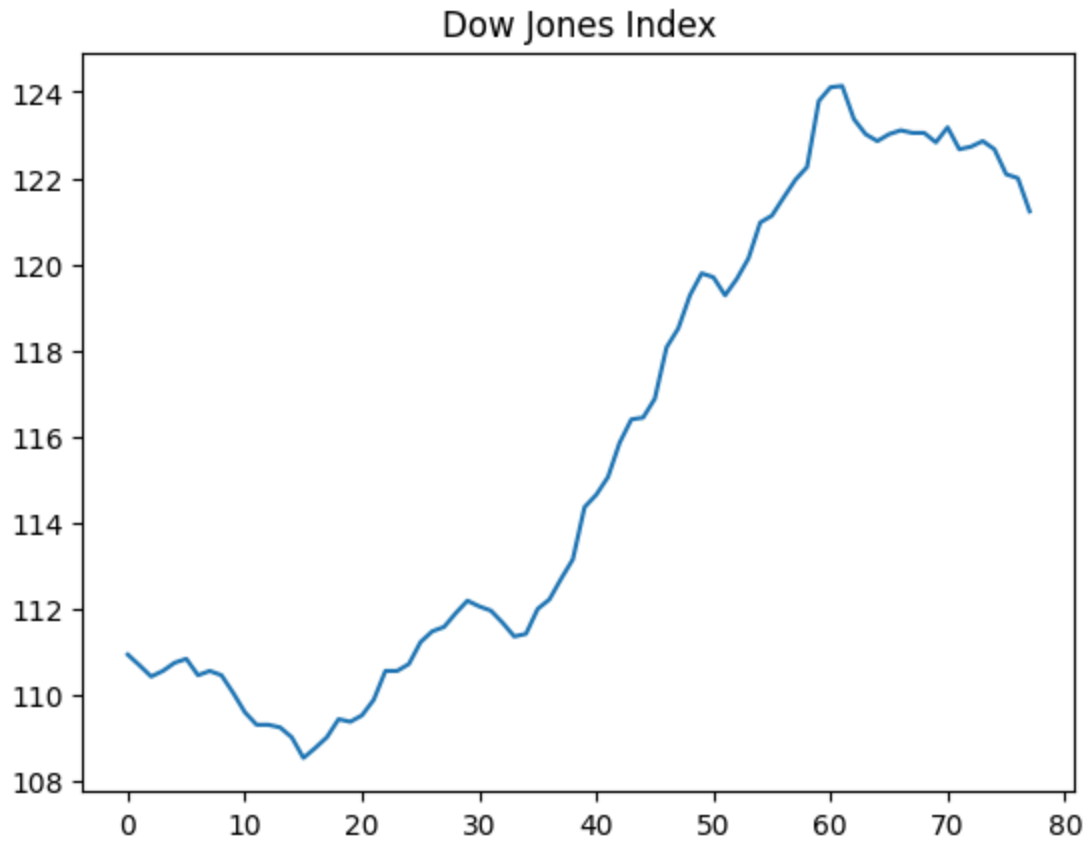
Model Selection and Forecasting with Financial Data

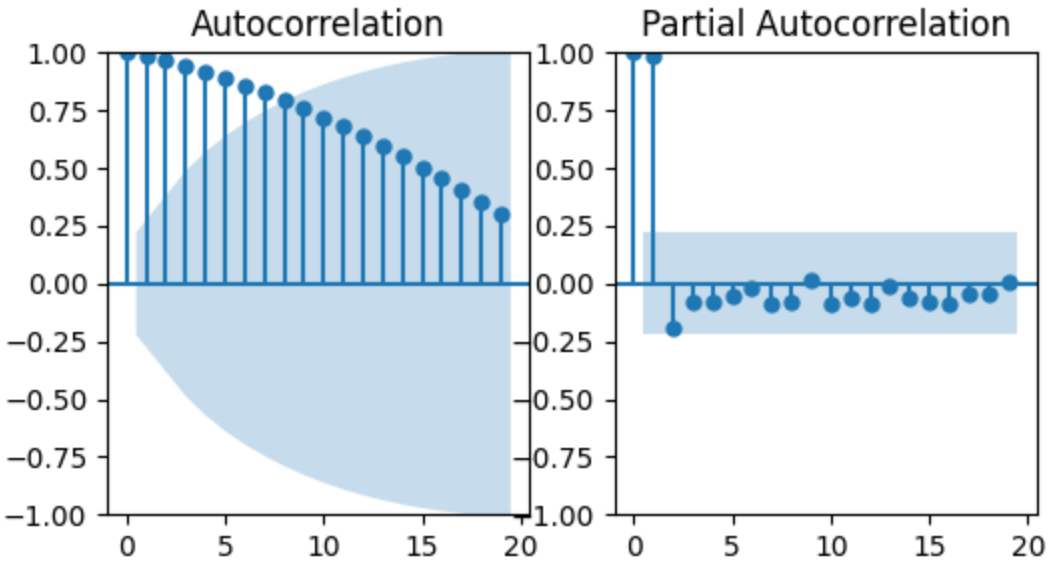
In [ ]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import warnings
warnings.filterwarnings('ignore')

dow = pd.read_csv('dowj.txt', sep=" ", header=None).rename({0: 'x'}, axis=1)
```

In [ ]:
```
plt.plot(dow)
plt.title('Dow Jones Index')
plt.show()
```



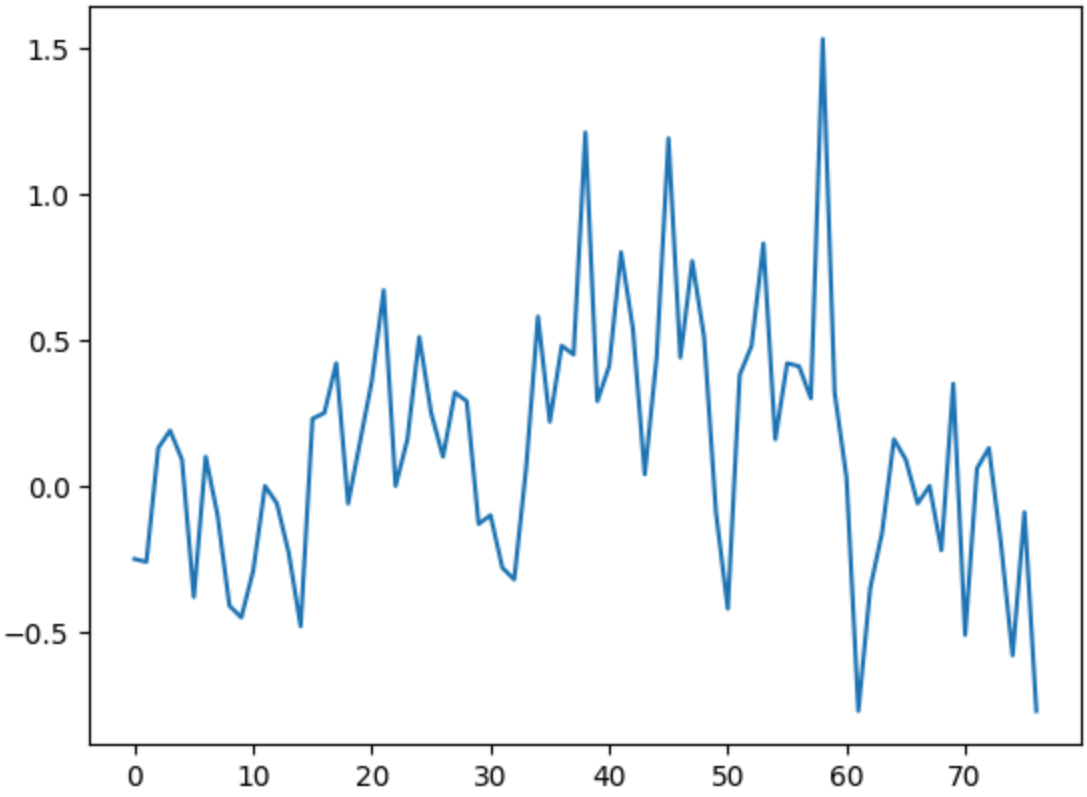
We can see a clear upward trend indicating non-stationarity. There is no clear stationarity at this point

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

fig, ax = plt.subplots(1,2,figsize=(6,3))

plot_acf(dow, ax=ax[0]);
plot_pacf(dow, ax=ax[1]);
```



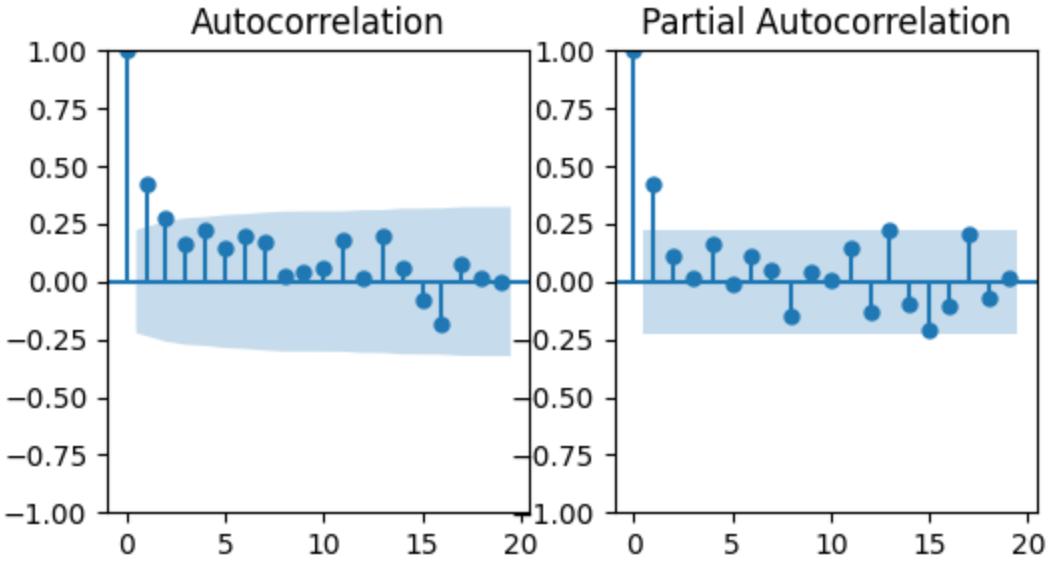
Make the data stationary

In [ ]:
```
d_dow = dow.diff()
d_dow = d_dow.dropna().reset_index(drop=True)
plt.plot(d_dow)
plt.show()
```



Updated ACF and PCAF

In [ ]:
```
fig, ax = plt.subplots(1,2, figsize=(6,3))
plot_acf(d_dow, ax=ax[0]);
plot_pacf(d_dow, ax=ax[1]);
```



We see evidenceof 1/2 MA lags and 1 AR lags.

In [ ]:
```
from statsmodels.regression.linear_model import yule_walker
from statsmodels.tsa.arima.model import ARIMA

ar_1 = yule_walker(d_dow, order=1)
print(f"Phi = {round(ar_1[0][0],3)}")
```

```
Phi = 0.427
```

AR(1) Parameter Estimation and CI

In [ ]:
```
model = ARIMA(d_dow, order=(1,0,0)).fit()
ci = model.conf_int(alpha=.05).loc['ar.L1']
print(f"Lower bound: {round(ci[0],3)}\nUpper bound: {round(ci[1],3)}")
```

```
Lower bound: 0.186
Upper bound: 0.709
```

AIC for ARMA p=q=0:6

In [ ]:
```
aic = pd.DataFrame(index=[f"p = {j}" for j in range(7)], columns=[f"q = {j}" for j in ra

for p_value in range(6):
    for q_value in range(6):
        model = ARIMA(d_dow,order=(p_value,0,q_value)).fit()
        aic.at[f"p = {p_value}", f"q = {q_value}"] = model.aic
```

In [ ]:
```
aic[aic == aic.min().min()]
```

Out[ ]:

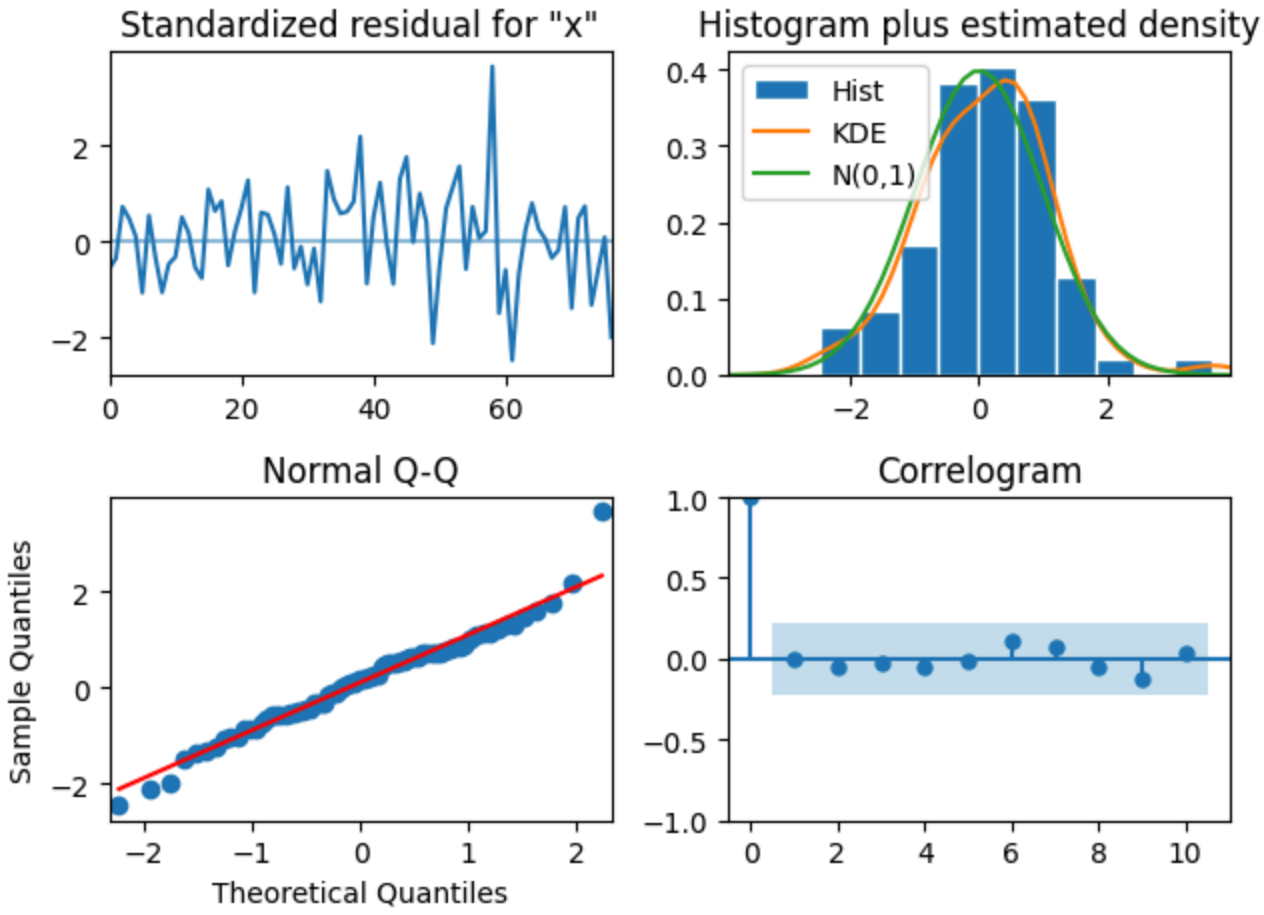
	q = 0	q = 1	q = 2	q = 3	q = 4	q = 5	q = 6
p = 0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
p = 1	NaN	NaN	NaN	NaN	NaN	NaN	NaN
p = 2	NaN	NaN	NaN	NaN	NaN	NaN	NaN
p = 3	NaN	NaN	NaN	NaN	NaN	NaN	NaN
p = 4	NaN	NaN	NaN	NaN	NaN	NaN	NaN
p = 5	NaN	NaN	73.695959	NaN	NaN	NaN	NaN
p = 6	NaN	NaN	NaN	NaN	NaN	NaN	NaN

The best performing model by AIC was ARMA(p=5,q=2).

Model Diagnostics

In [ ]:
```
model = ARIMA(dow, order=(5,1,2)).fit()
```

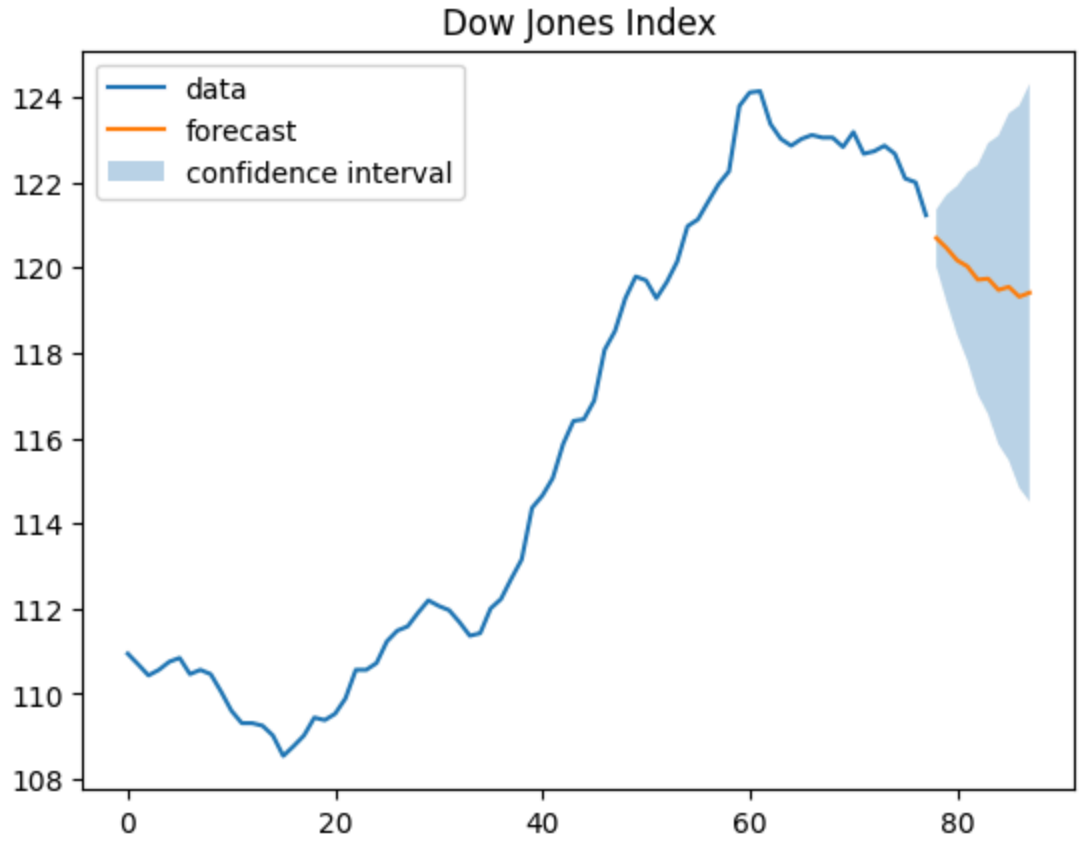
In [ ]:
```
fig = plt.figure()
model.plot_diagnostics(fig=fig);
fig.tight_layout()
```



The ARIMA(5,1,2) model performs quite well. The main concern is that the standarized residuals plot seems somewhat correlated. However, the residuals seem normally distributed, and the ACF plot shows no significant correlation.

Data Forecasting

In [ ]:
```
forecast_object = model.get_forecast(steps=10)
mean_forecast = forecast_object.predicted_mean
lower_bound = forecast_object.conf_int()['lower x']
upper_bound = forecast_object.conf_int()['upper x']

plt.plot(dow.index, dow, label='data')
plt.plot(mean_forecast.index, mean_forecast, label='forecast')
plt.fill_between(mean_forecast.index, lower_bound, upper_bound, alpha=.3, label='confide
plt.legend()
plt.title('Dow Jones Index')
plt.show()
```



In [ ]: