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

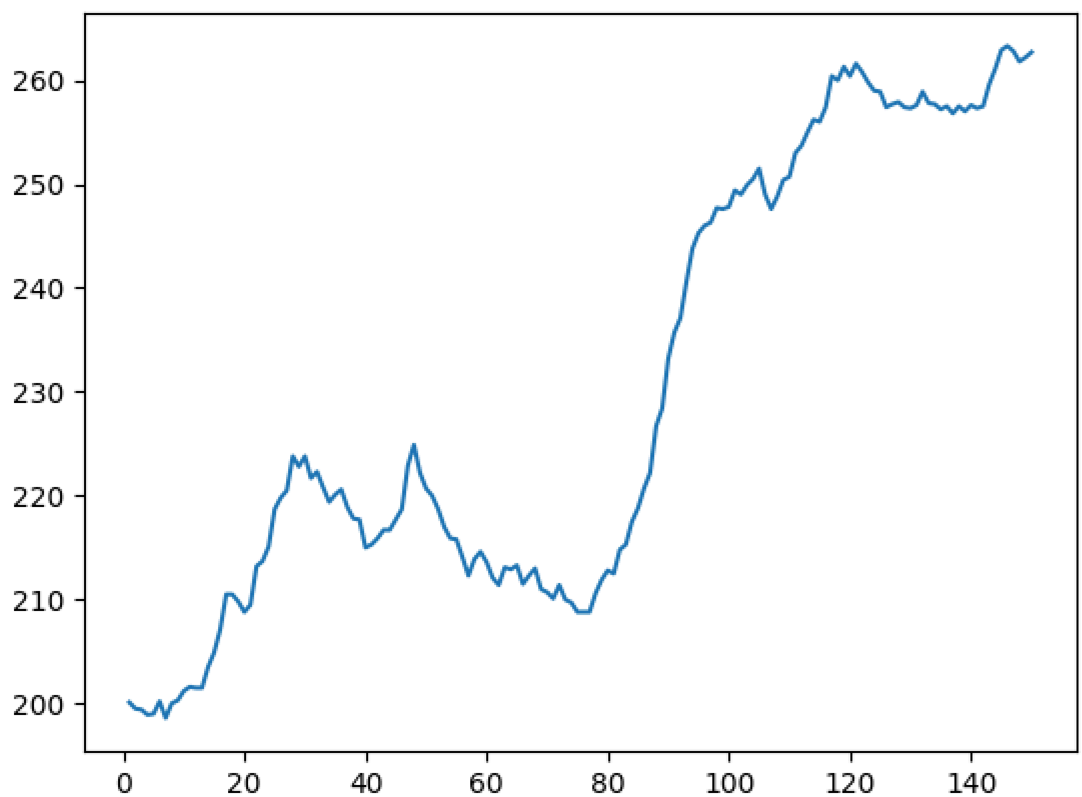
Problem 1

In [ ]:
```
data=pd.read_csv('airpassenger.csv', index_col=0)
```

Part a

We see evidence of a consistent, upward trend indicating a non-stationary process. However, the variance of the process seems mostly constant over time. At this point, there is not huge evidence of seasonality, but we will continue to look out for that.

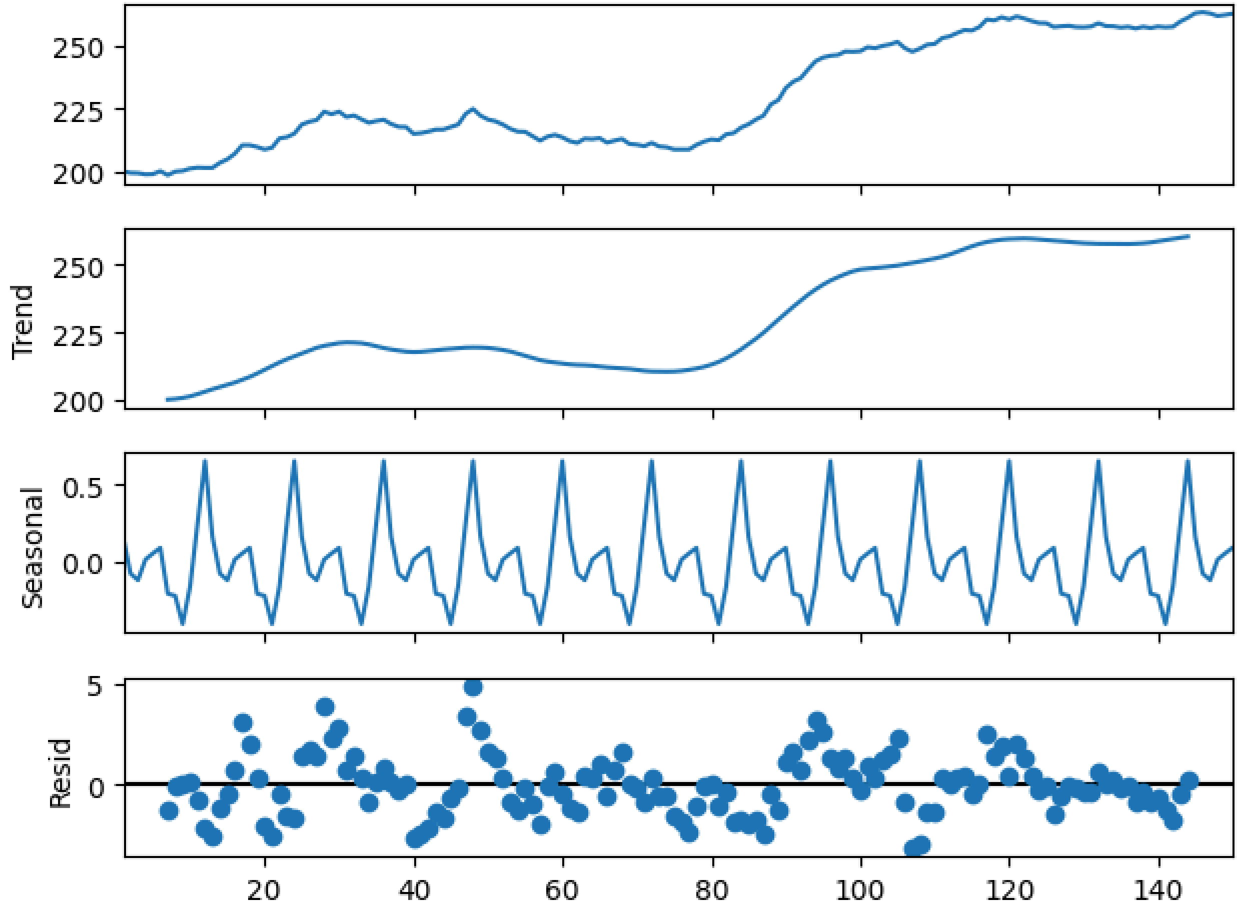
In [ ]:
```
plt.plot(data)
plt.show()
```



Part b

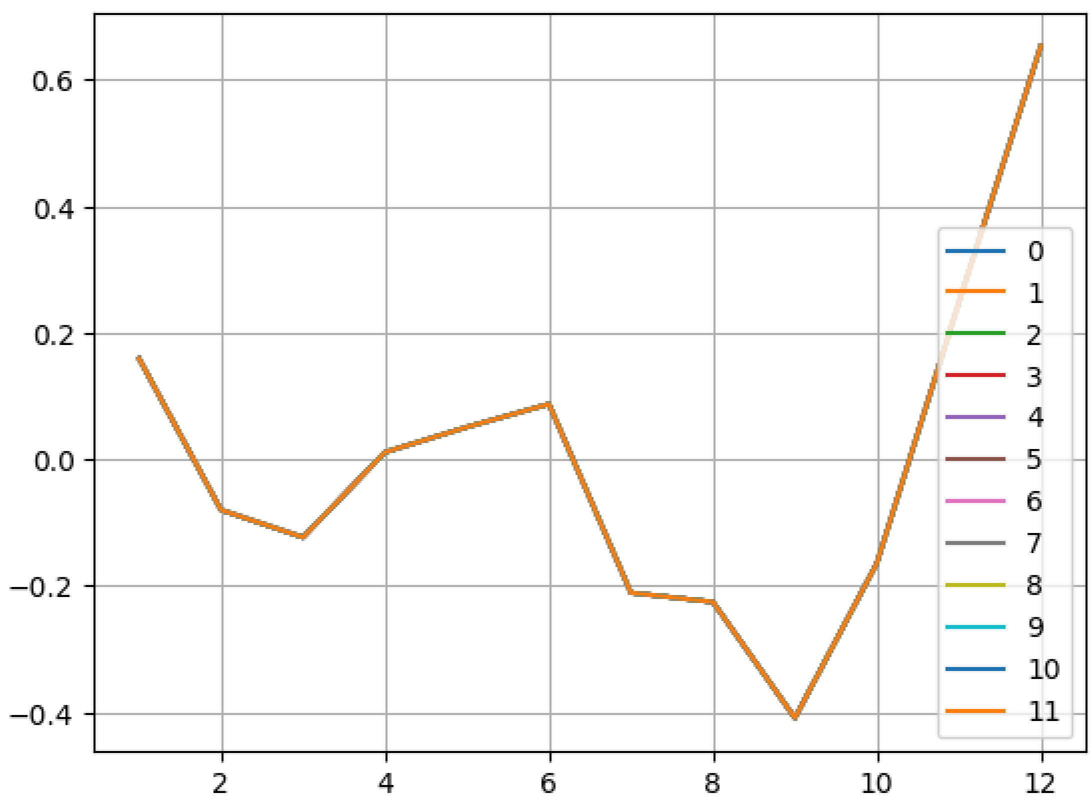
Now we see that there is evidence of seasonality with a period of 12 months.

In [ ]:
```
from statsmodels.tsa.seasonal import seasonal_decompose
result = seasonal_decompose(data, period=12)
result.plot();
```



In [ ]:
```
for j in range(12):
    plt.plot(range(1,13), result.seasonal[j*12:(j+1)*12], label=j)

plt.legend()
plt.grid()
plt.show()
```



It seems that the decompose function overfit the seasonal components, since they are all identical. This is also evident from the above decomposition plot.

Part c

In [ ]:

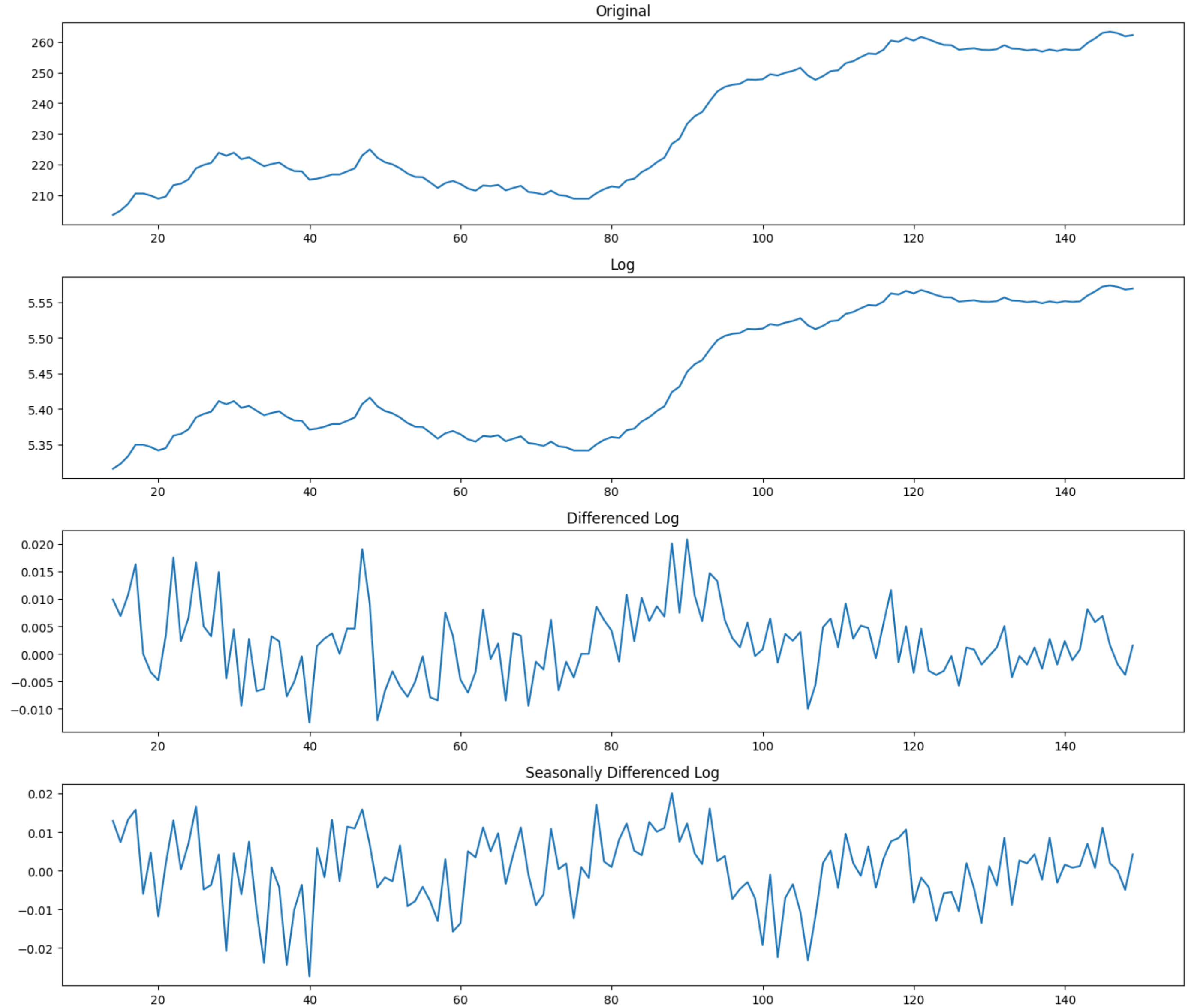
```
# Log Transform
log_data = np.log(data)
# Lag 1 differenced log data
dlog_data = log_data.diff()
# Lag 12 differenced lag 1 differenced log data
ddlog_data = dlog_data.diff(12)

combined_data = pd.DataFrame({
    'Original': data['x'],
    'Log': log_data['x'],
    'Differenced Log': dlog_data['x'],
    'Seasonally Differenced Log': ddlog_data['x']
}, index=range(len(data)))

# Drop NaN values that result from differencing
combined_data.dropna(inplace=True)

# Plotting
plt.figure(figsize=(14, 12))
for i, column in enumerate(combined_data.columns):
    plt.subplot(len(combined_data.columns), 1, i+1)
    plt.plot(combined_data.index, combined_data[column])
    plt.title(column)
    plt.tight_layout()

plt.show()
```

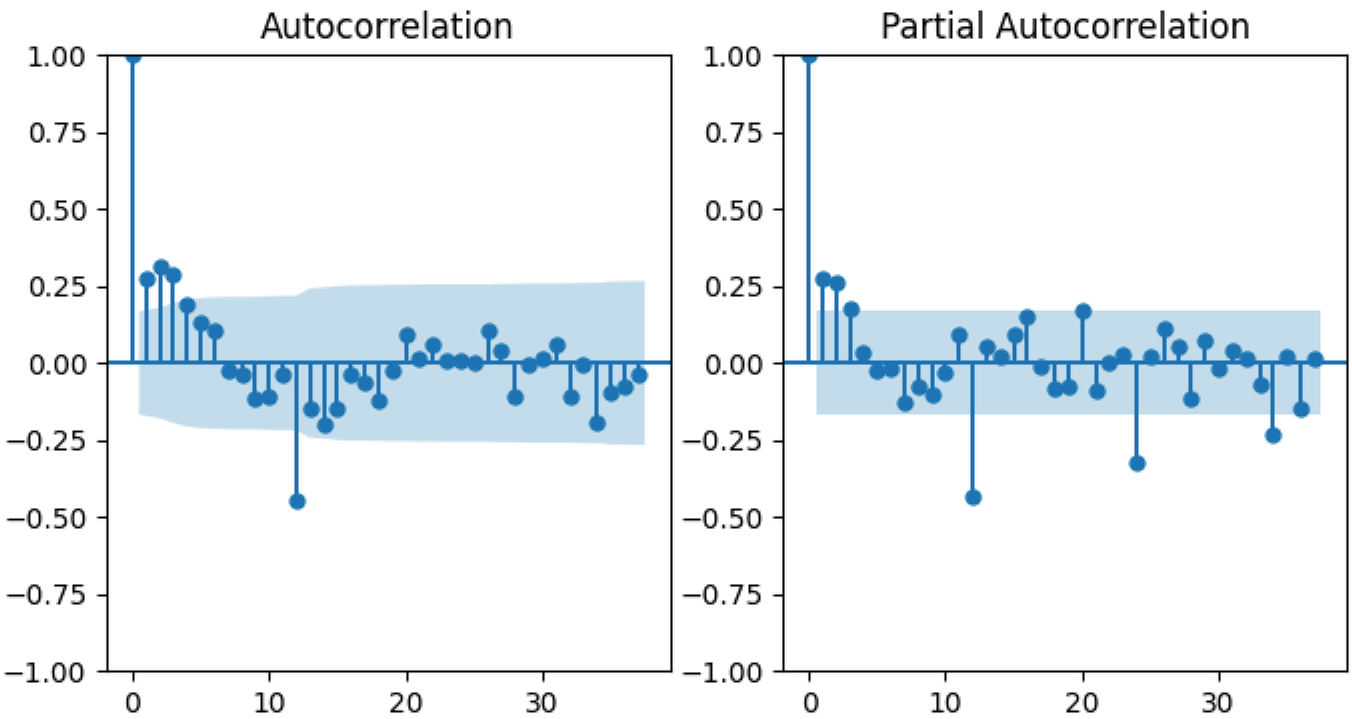


Part d

We see that the ACF function has significant spikes at 1,2 and 3, while the PACF plot has significant spikes at 1 and 2.

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

fig, ax = plt.subplots(1,2, figsize=(8,4))
plot_acf(ddlog_data.dropna(), ax=ax[0],lags=37);
plot_pacf(ddlog_data.dropna(), ax=ax[1], lags=37);
```

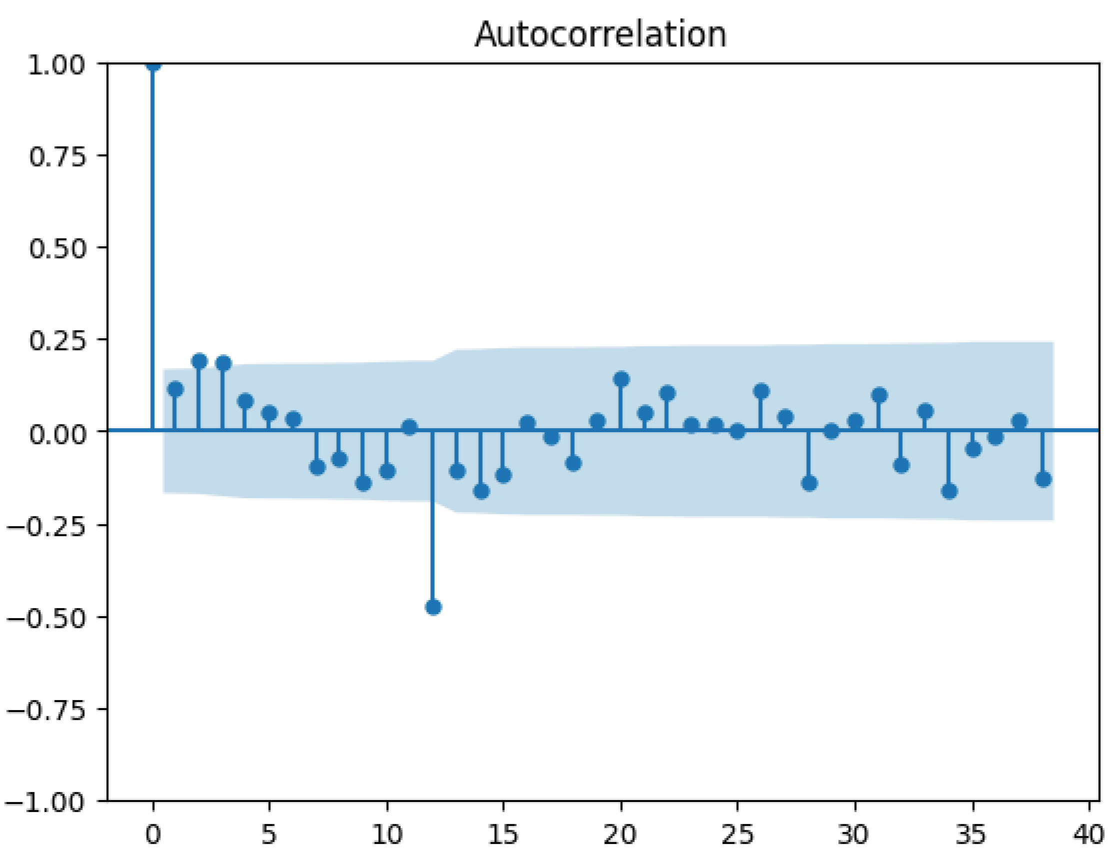


Part e

From the plot below, the residuals of the ARMA(1,1) seems to be mostly uncorrelated, except for at a lag of 12.

In [ ]:
```
from statsmodels.tsa.statespace.sarimax import SARIMAX

model = SARIMAX(ddlog_data.dropna().reset_index(drop=True), order=(1,0,1)).fit(disp=0);
plot_acf(model.resid, lags=38);
```



Part f

From the plot in part d, I will try the following models:

Model 1:

- (p,d,q) = (2,1,3)
- (P,D,Q)s = (0,1,1)12

Model 2:

- (p,d,q) = (0,1,0)
- (P,D,Q)s = (0,1,1)12

The difference between them is whether to include the the the p and q parameters.

Model 1

```
In [ ]: model_1 = SARIMAX(ddlog_data.dropna().reset_index(drop=True), order=(2,0,3), seasonal_or
        model_1.summary()
```

Out[ ]:

SARIMAX Results

Dep. Variable:	x	**No. Observations:**	137
Model:	SARIMAX(2, 0, 3)x(0, 0, [1], 12)	**Log Likelihood**	491.170
Date:	Tue, 05 Mar 2024	**AIC**	-968.341
Time:	19:42:31	**BIC**	-947.901
Sample:	0	**HQIC**	-960.035
	- 137		
Covariance Type:	opg		

	coef	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
ar.L1	0.9505	0.835	1.138	0.255	-0.687	2.588
ar.L2	-0.1888	0.693	-0.272	0.785	-1.548	1.170
ma.L1	-0.7559	0.840	-0.900	0.368	-2.402	0.890
ma.L2	0.1847	0.542	0.341	0.733	-0.878	1.247
ma.L3	0.0826	0.119	0.697	0.486	-0.150	0.315
ma.S.L12	-0.9536	0.439	-2.171	0.030	-1.815	-0.093
sigma2	3.717e-05	1.49e-05	2.492	0.013	7.94e-06	6.64e-05

Ljung-Box (L1) (Q):	0.00	**Jarque-Bera (JB):**	0.25
Prob(Q):	0.97	**Prob(JB):**	0.88
Heteroskedasticity (H):	0.48	**Skew:**	-0.10
Prob(H) (two-sided):	0.01	**Kurtosis:**	3.09

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

i) We see that only the ma.S.L12 parameter (Q) is significant. ii) The residuals seem stationary since they resemble white noise.

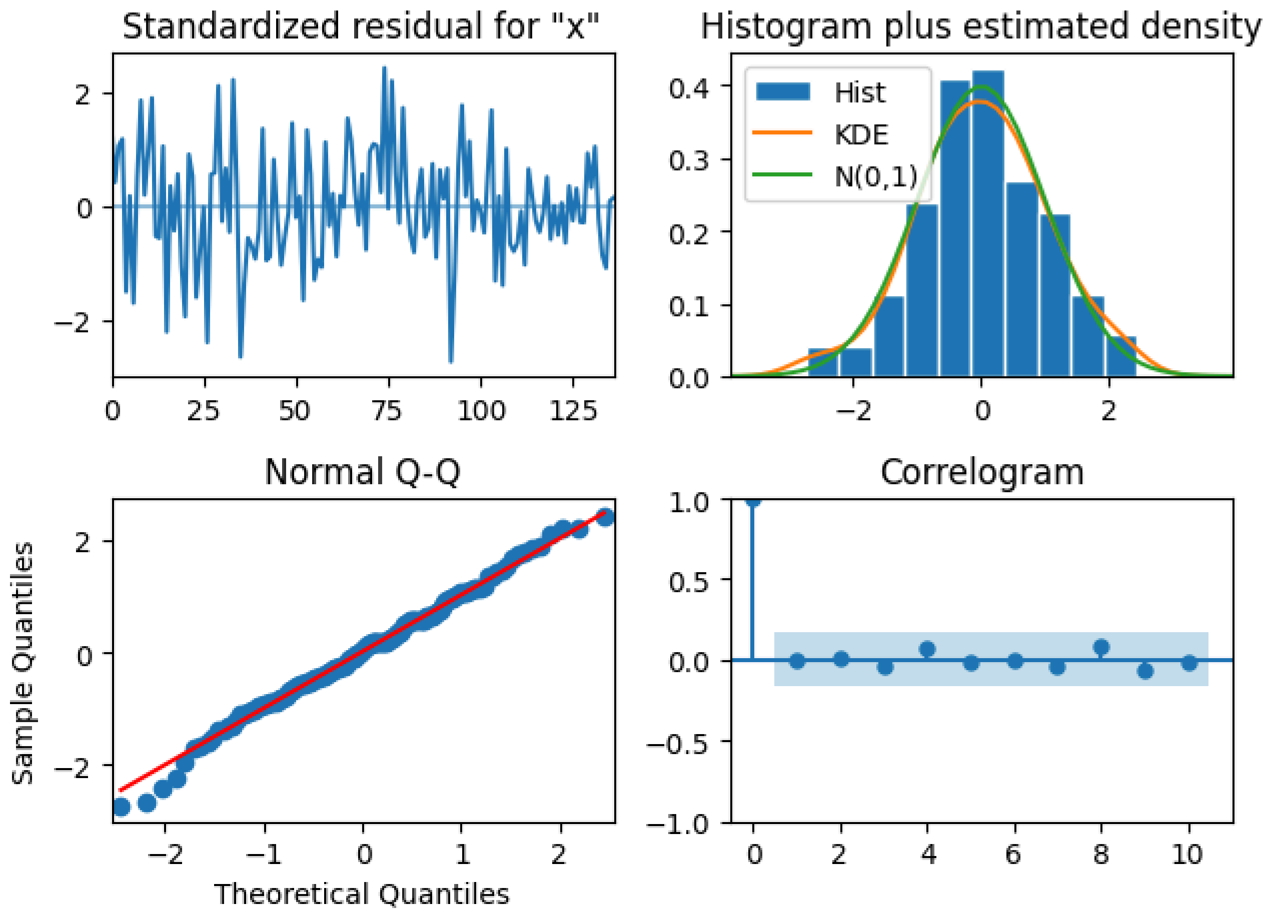
iii) The residuals are uncorrelated, indicating that our model captures most of the autocorrelation in the data.

iv) Yes, the standardized residuals seem to follow a gaussian distribution.

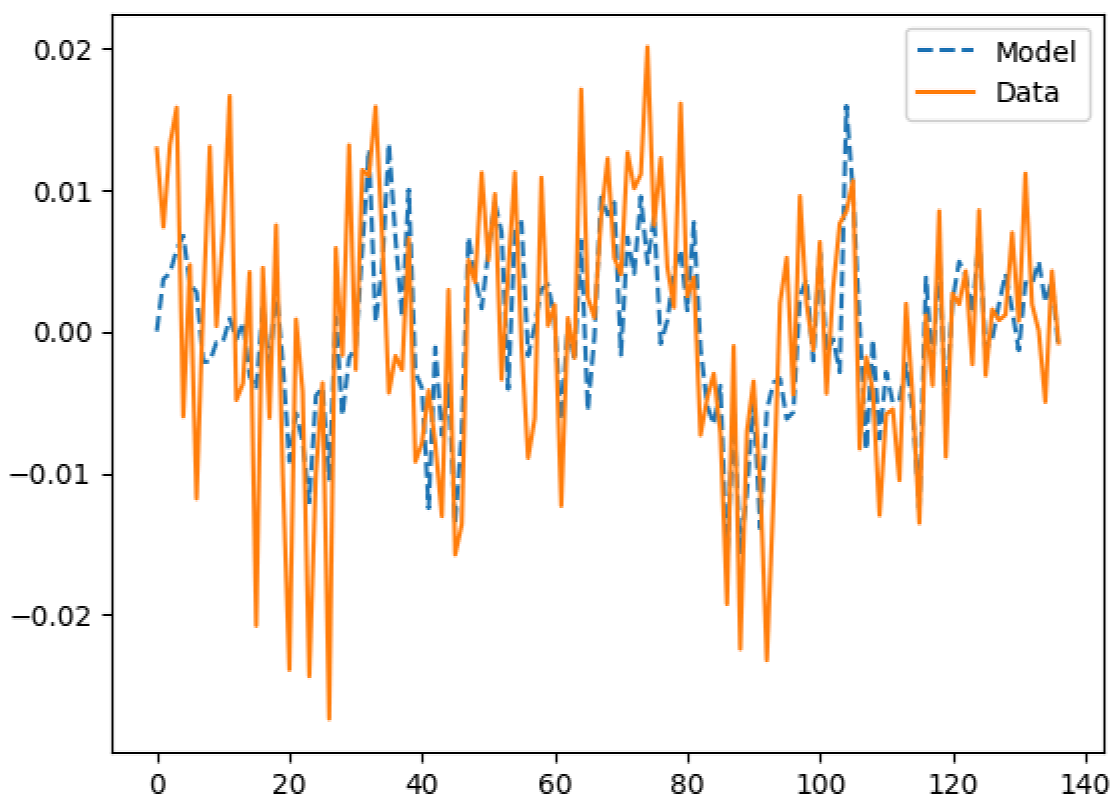
v) We accept Ho that residuals are WN since p=.99

In [ ]:

```
fig = plt.figure()
model_1.plot_diagnostics(fig=fig);
fig.tight_layout()
```



In [ ]:
```
plt.plot(model_1.fittedvalues, label='Model', ls='--')
plt.plot(ddlog_data.dropna().reset_index(drop=True), label='Data')
plt.legend()
plt.show()
```



Model 2

In [ ]:
```
model_2 = SARIMAX(ddlog_data.dropna().reset_index(drop=True), order=(0,0,0), seasonal_or
model_2.summary()
```

Out[ ]:

SARIMAX Results

Dep. Variable:	x	**No. Observations:**	137
Model:	SARIMAX(0, 0, [1], 12)	**Log Likelihood**	479.061
Date:	Tue, 05 Mar 2024	**AIC**	-954.121
Time:	19:43:01	**BIC**	-948.281
Sample:	0	**HQIC**	-951.748
	- 137		
Covariance Type:	opg		

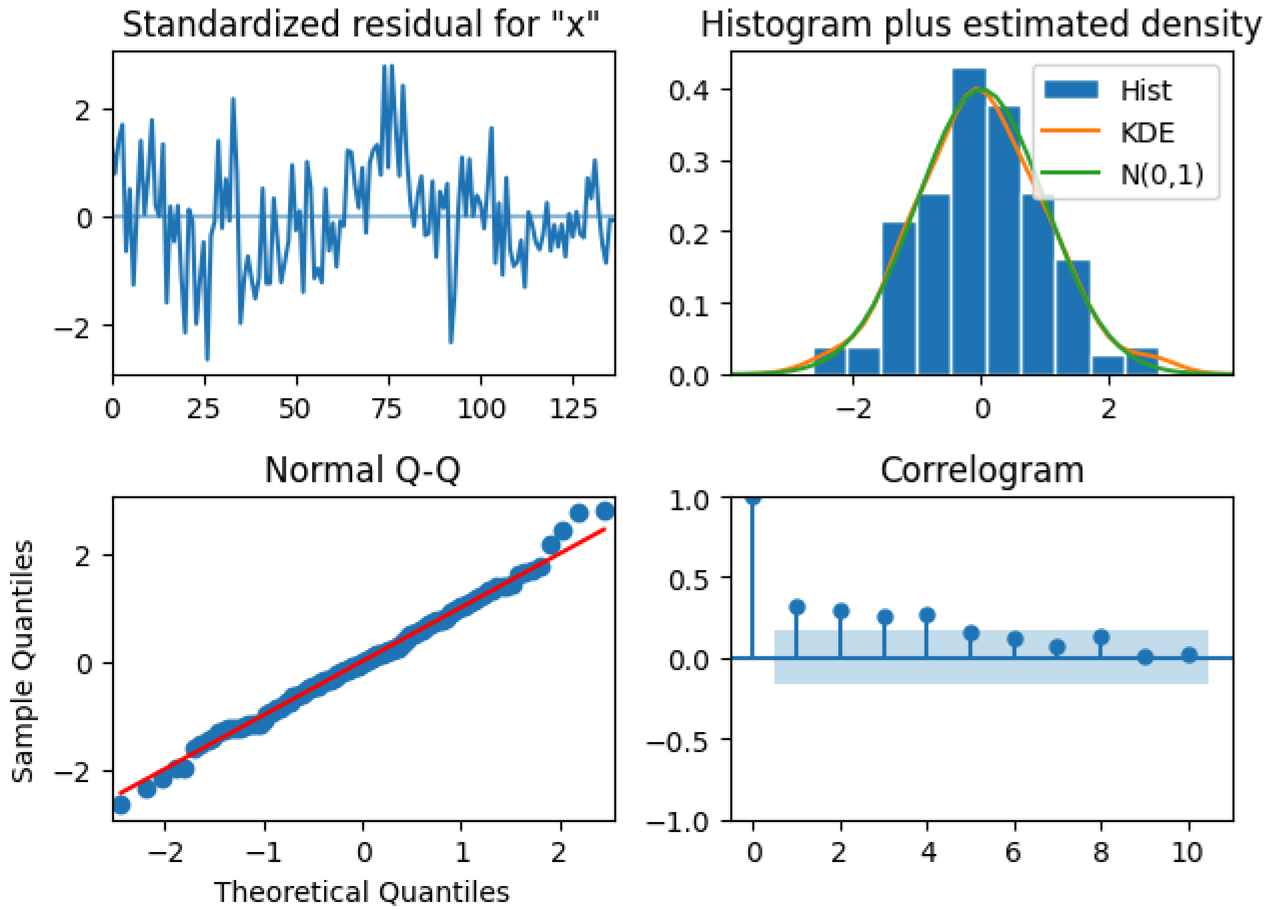
	coef	std err	z	P>|z|	[0.025	0.975]
ma.S.L12	-0.9798	1.061	-0.923	0.356	-3.060	1.101
sigma2	4.444e-05	4.58e-05	0.970	0.332	-4.54e-05	0.000

Ljung-Box (L1) (Q):	13.92	**Jarque-Bera (JB):**	0.71
Prob(Q):	0.00	**Prob(JB):**	0.70
Heteroskedasticity (H):	0.41	**Skew:**	0.14
Prob(H) (two-sided):	0.00	**Kurtosis:**	3.22

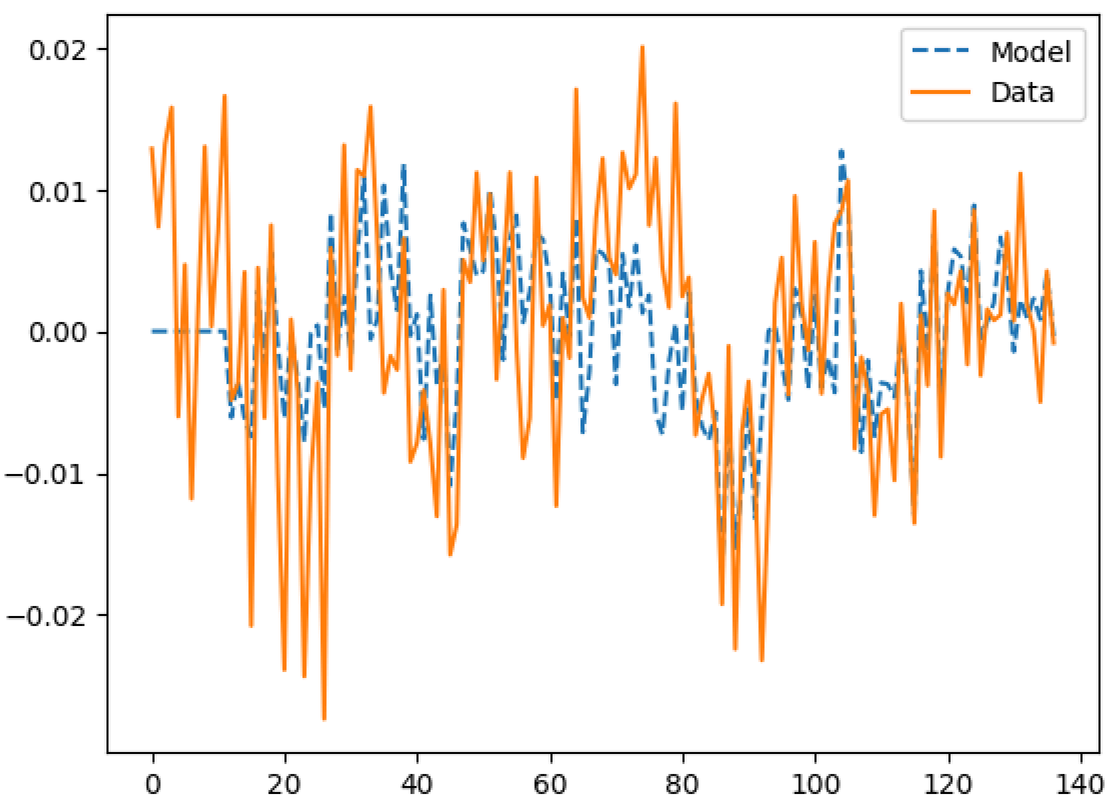
Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).

i) No parameters are significant.
ii) The plot of the residuals looks similar to White Noise but less random/more correlation.
iii) We can see that some residuals are correlated.
iv) The standarized residuals seem to be normally distributed.
v) We must reject the Ho that the residuals are WN b/c p=0.00.

In [ ]:
```
fig = plt.figure()
model_2.plot_diagnostics(fig=fig);
fig.tight_layout()
```



In [ ]:
```
plt.plot(model_2.fittedvalues, label='Model', ls='--')
plt.plot(ddlog_data.dropna().reset_index(drop=True), label='Data')
plt.legend()
plt.show()
```

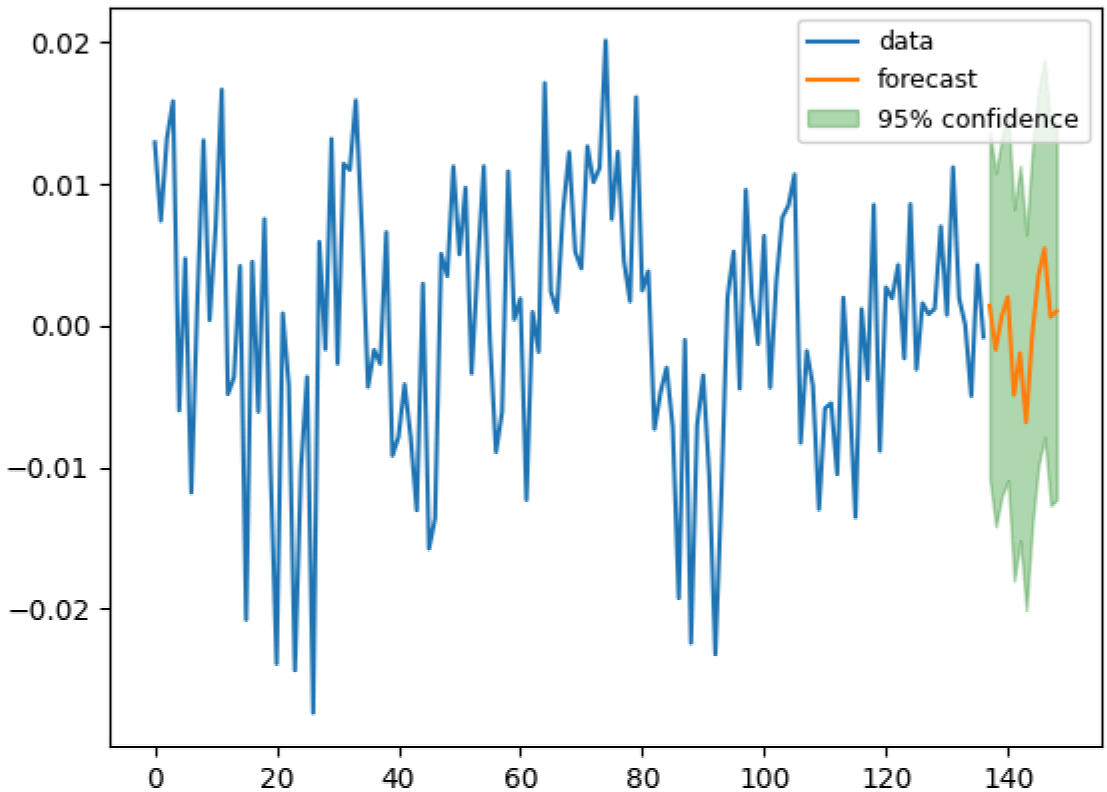


vi) The second model has the lowest information criterion, however, it is important to consider that it failed the Ljung-Box text, indicating that its residuals are not WN and likely correlated.

Part g

In [ ]:
```
forecast_object = model_1.get_forecast(steps=12)
forecast = forecast_object.predicted_mean
lower_bound = forecast_object.conf_int()['lower x']
upper_bound = forecast_object.conf_int()['upper x']

plt.plot(ddlog_data.dropna().reset_index(drop=True).index, ddlog_data.dropna().reset_ind
plt.plot(forecast.index, forecast, label='forecast')
plt.fill_between(forecast.index, lower_bound, upper_bound, color='green', alpha=0.3, lab
plt.legend(fontsize=9)
plt.show()
```



Yes the forecast seems reasonable based on recent trends.

Part h

Model 1:

$$\Phi(B^{12})\,\phi(B)\,(1-B^{12})(1-B)\,X_t = \Theta(B^{12})\,\theta(B)\,Z_t$$

Where

$$\Phi(B^{12}) = 1$$

$$\phi(B) = (1 - B\phi_1 - B^2\phi_2)$$

$$\Theta(B^{12}) = (1 - \Theta_1 B^{12})$$

$$\theta(B) = (1 + \theta_1 B + \theta_2 B^2)$$

$$\boxed{(1 - B\phi_1 - B^2\phi_2)(1 - B^{12})(1-B)X_t = (1 - \Theta_1 B^{12}) \cdot (1 + \theta_1 B + \theta_2 B^2) Z_t}$$