ARIMA Model

Date: Fri Aug 5 10:01:47 2022

Author: kkitonga and ewayagi

In [1]:
```
#==================================libraries================================#

import numpy as np
import pandas as pd
import statsmodels.api as sm
import matplotlib.pyplot as plt
```

In [2]:
```
#=======================loading the data from R package====================#

arimadata = sm.datasets.get_rdataset("GoldSilver", "AER").data

arima_index = arimadata.reset_index()
print(arima_index)

arima_reset = arima_index.rename(columns = {"index" : "date"})
print(arima_reset)
```

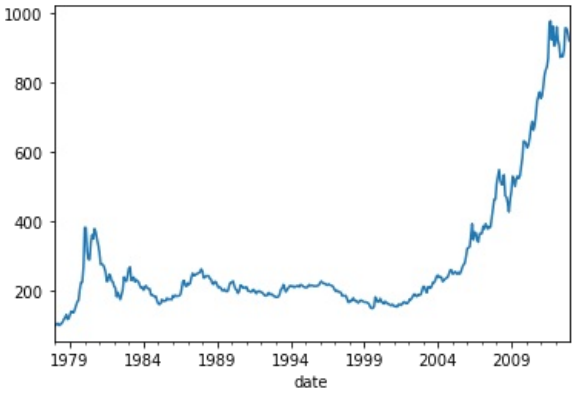
```
           index    gold   silver
0     1977-12-30  100.00   223.42
1     1978-01-02  100.00   223.42
2     1978-01-03  100.00   229.84
3     1978-01-04  100.00   224.58
4     1978-01-05  100.00   227.99
...          ...     ...      ...
9127  2012-12-25  906.96  1088.32
9128  2012-12-26  907.61  1093.34
9129  2012-12-27  909.26  1100.81
9130  2012-12-28  905.00  1091.16
9131  2012-12-31  915.88  1100.34

[9132 rows x 3 columns]
            date    gold   silver
0     1977-12-30  100.00   223.42
1     1978-01-02  100.00   223.42
2     1978-01-03  100.00   229.84
3     1978-01-04  100.00   224.58
4     1978-01-05  100.00   227.99
...          ...     ...      ...
9127  2012-12-25  906.96  1088.32
9128  2012-12-26  907.61  1093.34
9129  2012-12-27  909.26  1100.81
9130  2012-12-28  905.00  1091.16
9131  2012-12-31  915.88  1100.34

[9132 rows x 3 columns]
```

In [3]:
```
#=================Resampling the data to monthly prices====================#

arima_reset['date'] = pd.to_datetime(arima_reset['date'])
arima_reset = arima_reset.set_index('date')
monthly_prices = arima_reset.resample('M').mean()
print(monthly_prices)
```

```
                gold       silver
date
1977-12-31  100.000000   223.420000
1978-01-31  101.786364   229.861818
1978-02-28  104.397000   230.133000
1978-03-31  107.171739   245.419130
1978-04-30  101.579500   237.242000
...                ...          ...
2012-08-31  893.377391  1058.241304
2012-09-30  956.595500  1234.286000
2012-10-31  955.360870  1211.120435
2012-11-30  942.341364  1197.614545
2012-12-31  920.527143  1154.132381

[421 rows x 2 columns]
```

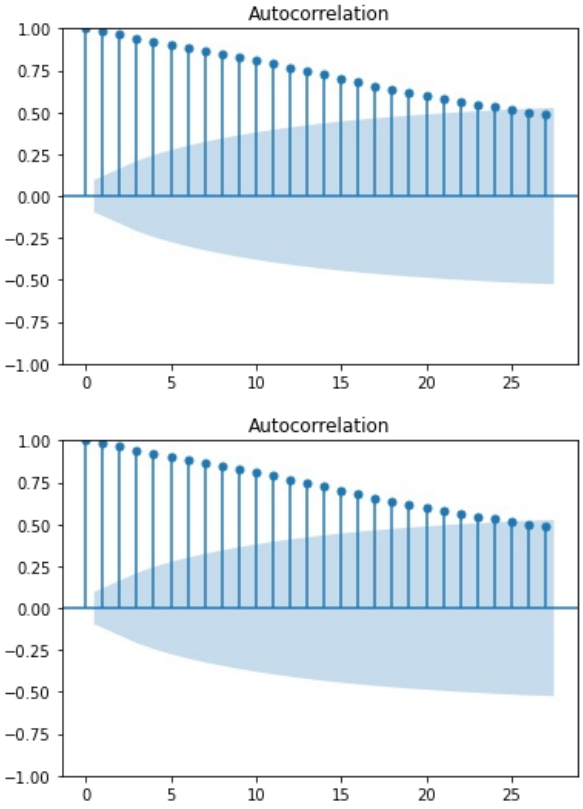
In [4]:
```
#================================plots=====================================#

arimagold = monthly_prices

arimagold['gold'].plot()

# this is a non-stationary series
```

Out[4]: <AxesSubplot:xlabel='date'>



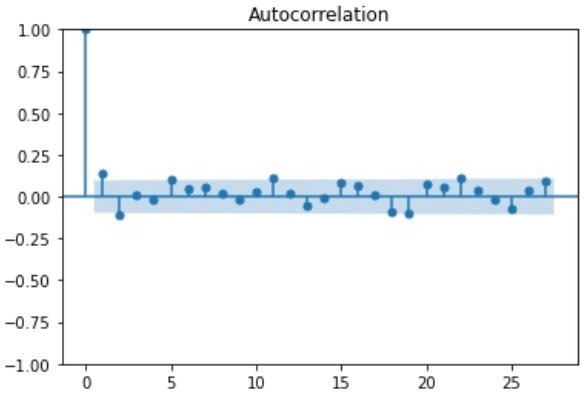
In [5]:
```
#=========================arima parameters; p, d, q============================#

from statsmodels.graphics.tsaplots import plot_acf

plot_acf(arimagold.gold)
```

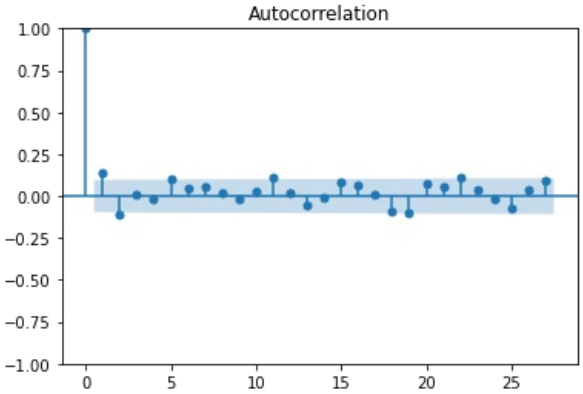
Out[5]:

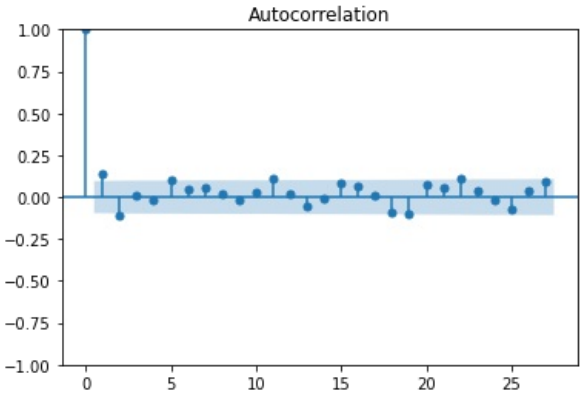


In [6]:
```
#=======================determining the order of differencing, d==============#

plot_acf(arimagold.gold.diff()[1:])

plot_acf(arimagold.gold.diff()[2:])

# with first and second order differencing, there is no autocorrelation
```

Out[6]:







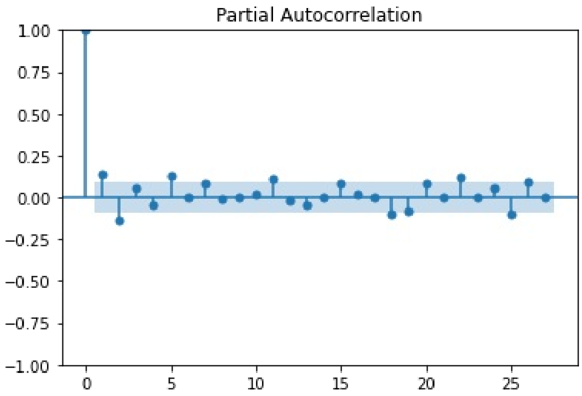
In [7]:
```
#======================unit root or stationarity test=========================#

from statsmodels.tsa.stattools import adfuller

adftest1 = adfuller(arimagold['gold'])
print(adftest1[0])
print(adftest1[1])

adftest2 = adfuller(arimagold['gold'].diff()[1:])
print(adftest2[0])
print(adftest2[1])

adftest3 = adfuller(arimagold['gold'].diff()[2:])
print(adftest3[0])
print(adftest3[1])

# conclusion: choose d = 1
```

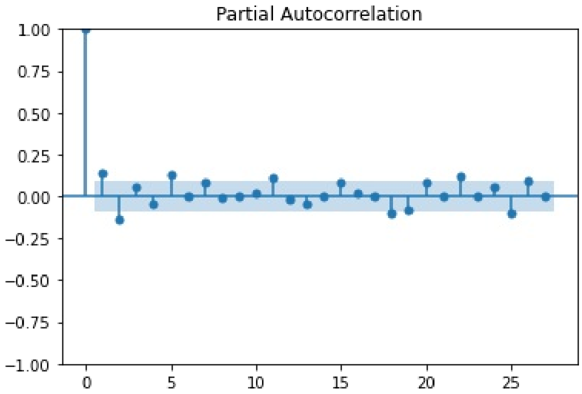
```
1.7102073619071072
0.9981532615440724
-8.12419570286498
1.1347189754713644e-12
-8.11347406815153
1.2083171395117218e-12
```

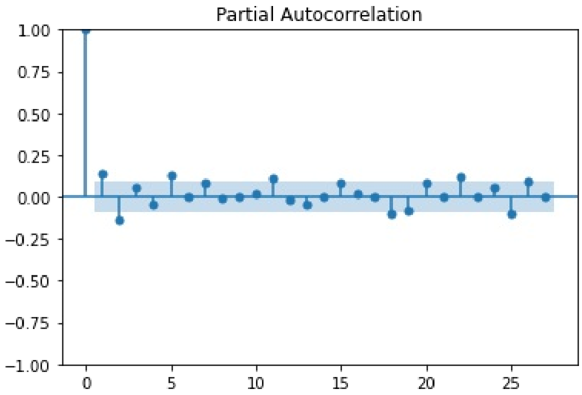
In [8]:
```
#==============deterining the order of the autoregressive model, p============#

from statsmodels.graphics.tsaplots import plot_pacf

plot_pacf(arimagold.gold.diff()[1:])
plot_pacf(arimagold.gold.diff()[2:])

#conclusion: choose p = 1;
#let us use q = 2; from the ACF plots
```

```
C:\Users\ewaya\Anaconda3\lib\site-packages\statsmodels\graphics\tsaplots.py:348: FutureWarning: The default met
hod 'yw' can produce PACF values outside of the [-1,1] interval. After 0.13, the default will change tounadjust
ed Yule-Walker ('ywm'). You can use this method now by setting method='ywm'.
  warnings.warn(
C:\Users\ewaya\Anaconda3\lib\site-packages\statsmodels\graphics\tsaplots.py:348: FutureWarning: The default met
hod 'yw' can produce PACF values outside of the [-1,1] interval. After 0.13, the default will change tounadjust
ed Yule-Walker ('ywm'). You can use this method now by setting method='ywm'.
  warnings.warn(
```

Out[8]:





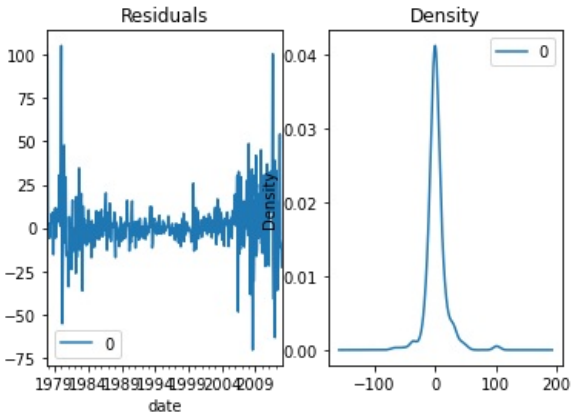


In [9]:

```
#====================================ARIMA model=============================#

from statsmodels.tsa.arima.model import ARIMA

arimamodel = ARIMA(arimagold.gold, order = (1, 1, 2))
arimamodel_fit = arimamodel.fit()
print(arimamodel_fit.summary())
```

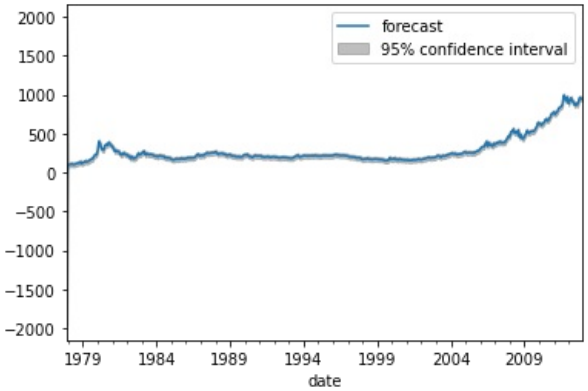
```
                               SARIMAX Results
==========================================================================================
Dep. Variable:                      gold   No. Observations:                  421
Model:                 ARIMA(1, 1, 2)   Log Likelihood               -1753.444
Date:                Mon, 05 Sep 2022   AIC                           3514.887
Time:                        05:22:36   BIC                           3531.048
Sample:                    12-31-1977   HQIC                          3521.275
                         - 12-31-2012
Covariance Type:                  opg
==========================================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.6031      0.105     -5.771      0.000      -0.808      -0.398
ma.L1          0.8031      0.111      7.254      0.000       0.586       1.020
ma.L2          0.0078      0.050      0.154      0.877      -0.091       0.106
sigma2       247.4557      7.351     33.664      0.000     233.048     261.863
===================================================================================
Ljung-Box (L1) (Q):                   0.06   Jarque-Bera (JB):              1946.72
Prob(Q):                              0.80   Prob(JB):                         0.00
Heteroskedasticity (H):               1.89   Skew:                             1.05
Prob(H) (two-sided):                  0.00   Kurtosis:                        13.34
===================================================================================

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

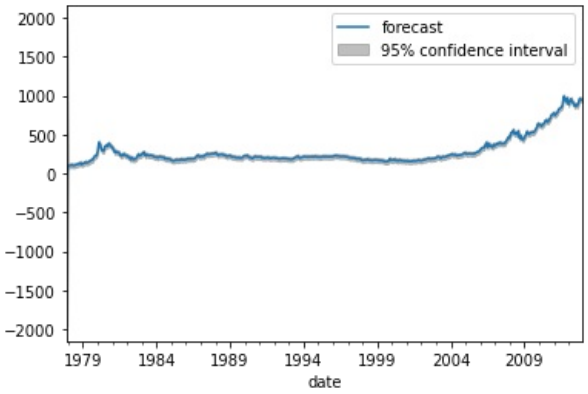
In [10]:
```
#==================ploting the ARIMA model=================================#

#plotting the residuals and density functions

resid = pd.DataFrame(arimamodel_fit.resid)
fig, ax = plt.subplots(1,2)
resid.plot(title = "Residuals", ax = ax[0])
resid.plot(kind = 'kde', title = 'Density', ax = ax[1])
plt.show()


#the plots shows a fair distribution of residual errors around them mean with a uniform variance
```



In [11]:
```
#============================predicted ARIMA model plot=======================#

from statsmodels.graphics.tsaplots import plot_predict

plot_predict(arimamodel_fit)
```

Out[11]:



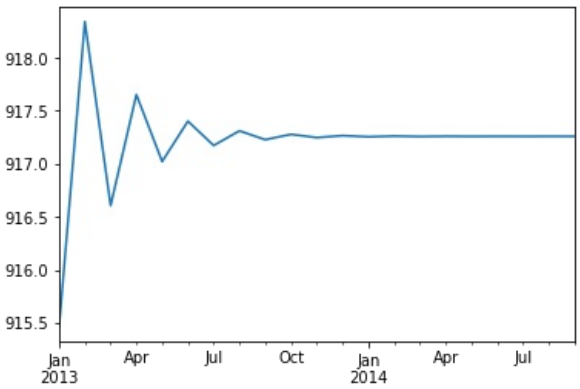


In [12]:
```
#==========predict the mean prices of gold for thr next 20 months=============#

predicted_values = arimamodel_fit.predict(start = len(arimagold),\
        end = len(arimagold) + 20, type = "levels").rename('predictions')
print(predicted_values)
```

```
2013-01-31    915.463992
2013-02-28    918.341977
2013-03-31    916.606249
2013-04-30    917.653076
2013-05-31    917.021729
2013-06-30    917.402498
2013-07-31    917.172854
2013-08-31    917.311353
2013-09-30    917.227824
2013-10-31    917.278201
2013-11-30    917.247818
2013-12-31    917.266142
2014-01-31    917.255091
2014-02-28    917.261756
2014-03-31    917.257736
2014-04-30    917.260161
2014-05-31    917.258698
2014-06-30    917.259580
2014-07-31    917.259048
2014-08-31    917.259369
2014-09-30    917.259176
Freq: M, Name: predictions, dtype: float64
```

In [13]:
```
#============================plot the predicted values=======================#

predicted_values.plot()
```

Out[13]:
```
<AxesSubplot:>
```



In [14]:
```
#=========================================THE END============================#
```

In [ ]: