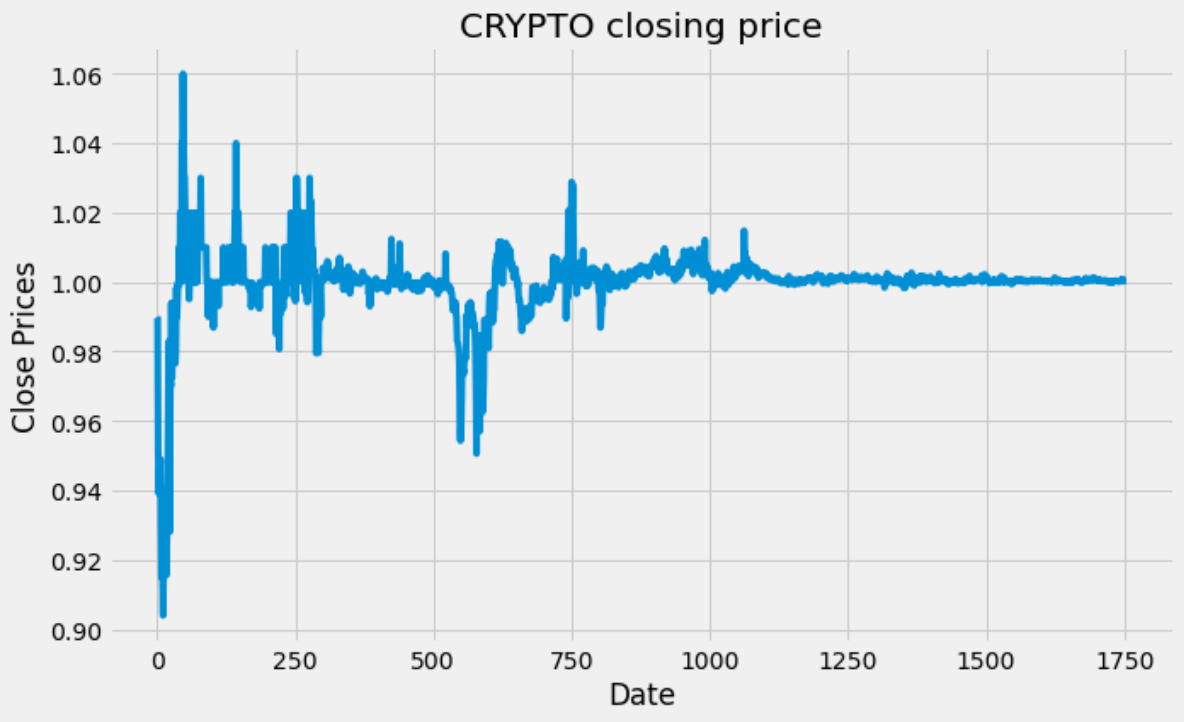
In [70]:

```
"""
https://www.researchgate.net/publication/346014059_Getting_to_a_feasible_income_equality
https://www.researchgate.net/publication/329631238_The_Anti-Social_System_Properties_Bitcoi
https://www.researchgate.net/publication/233688184_The_Gini_Index_and_Measures_of_Inequalit

"""

import warnings
warnings.filterwarnings('ignore')
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')
from pylab import rcParams
rcParams['figure.figsize'] = 10, 6
from statsmodels.tsa.stattools import adfuller
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.tsa.arima_model import ARIMA

from sklearn.metrics import mean_squared_error, mean_absolute_error
import math

df = pd.read_csv("Tether.csv");
print(df)
```

```
              Date   Price    Open    High     Low     Vol. Change %
0     Apr 14, 2017  0.9899  0.9939  0.9960  0.9689   71.65K   -0.40%
1     Apr 15, 2017  0.9896  0.9899  0.9899  0.9676   14.92K   -0.03%
2     Apr 16, 2017  0.9515  0.9896  0.9896  0.9515   12.41K   -3.85%
3     Apr 17, 2017  0.9392  0.9515  0.9698  0.9211   38.75K   -1.29%
4     Apr 18, 2017  0.9450  0.9392  0.9628  0.9207   39.25K    0.62%
...            ...     ...     ...     ...     ...      ...      ...
1749  Jan 28, 2022  1.0004  1.0002  1.0008  1.0002  198.11M    0.02%
1750  Jan 29, 2022  1.0004  1.0004  1.0007  1.0004   70.11M    0.00%
1751  Jan 30, 2022  1.0005  1.0003  1.0007  1.0003   42.17M    0.01%
1752  Jan 31, 2022  1.0005  1.0005  1.0008  1.0003  177.59M    0.00%
1753  Feb 01, 2022  1.0005  1.0005  1.0008  1.0004   85.69M    0.00%

[1754 rows x 7 columns]
```

In [73]:

```
'''df['Price'] = df['Price'].str.replace(',', '').astype(float)'''
plt.figure(figsize=(10,6))
plt.grid(True)
plt.xlabel('Date')
plt.ylabel('Close Prices')
plt.plot(df['Price'])
plt.title('CRYPTO closing price')
plt.show()
```

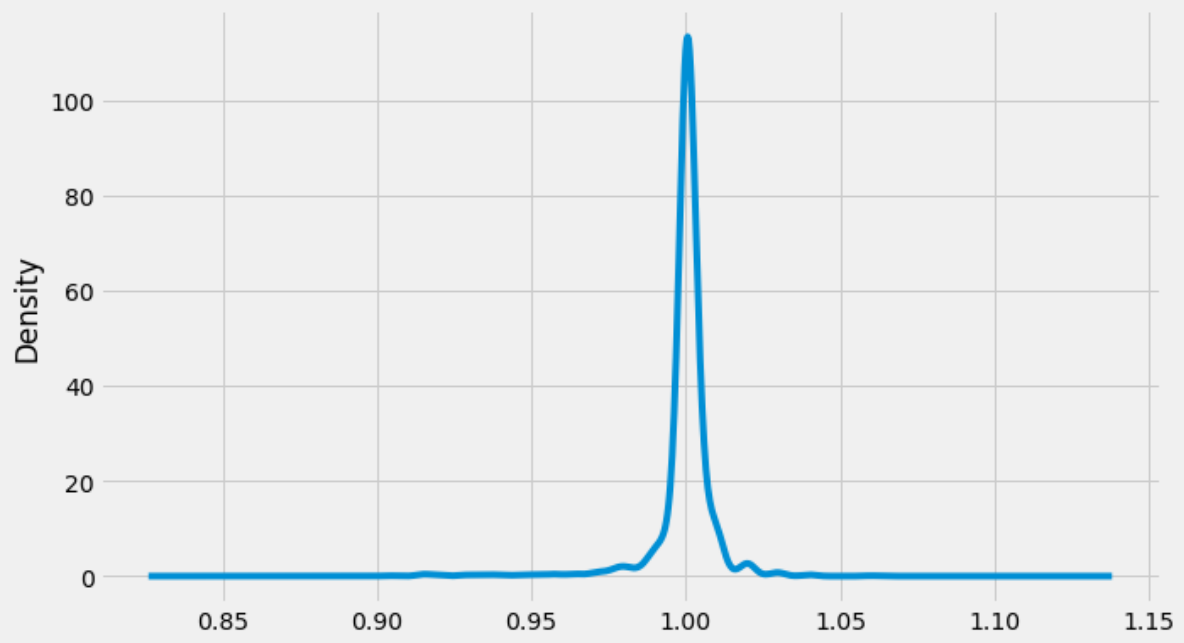


In [74]:

```
df['Price'].plot(kind='kde')
```

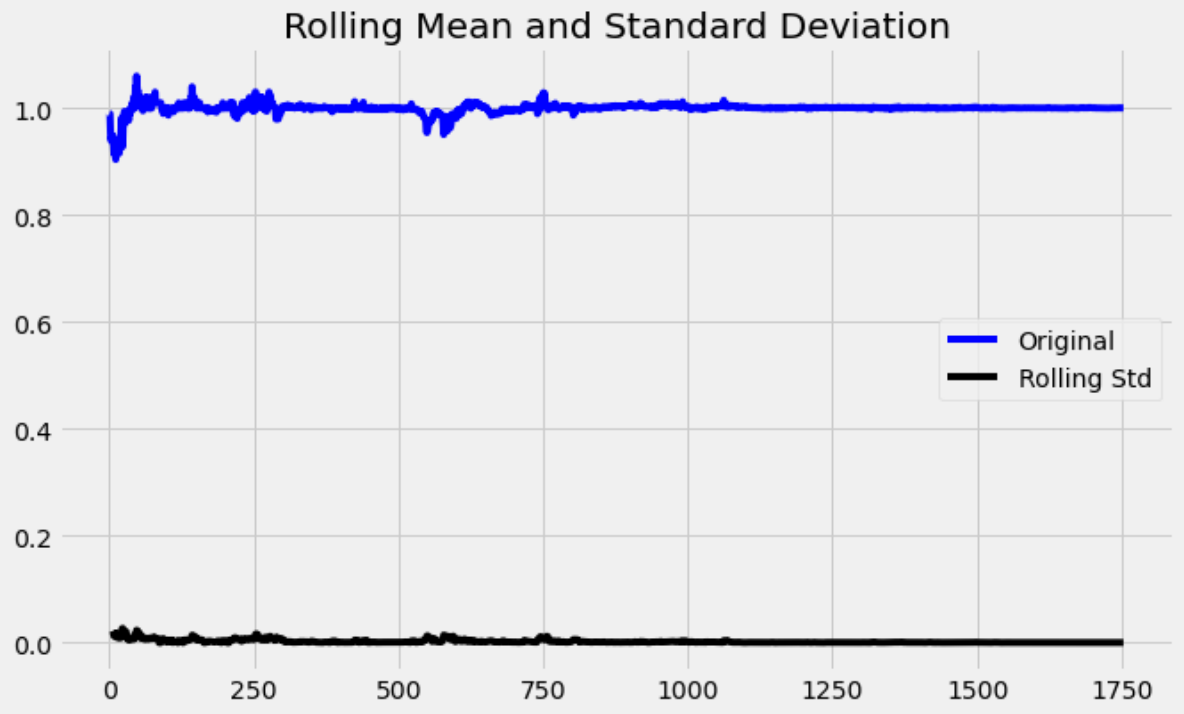
Out[74]:

```
<AxesSubplot:ylabel='Density'>
```



In [75]:

```
rolstd = df['Price'].rolling(7).std()
plt.plot(df['Price'], color='blue',label='Original')
plt.plot(rolstd, color='black', label = 'Rolling Std')
plt.legend(loc='best')
plt.title('Rolling Mean and Standard Deviation')
plt.show(block=False)
```

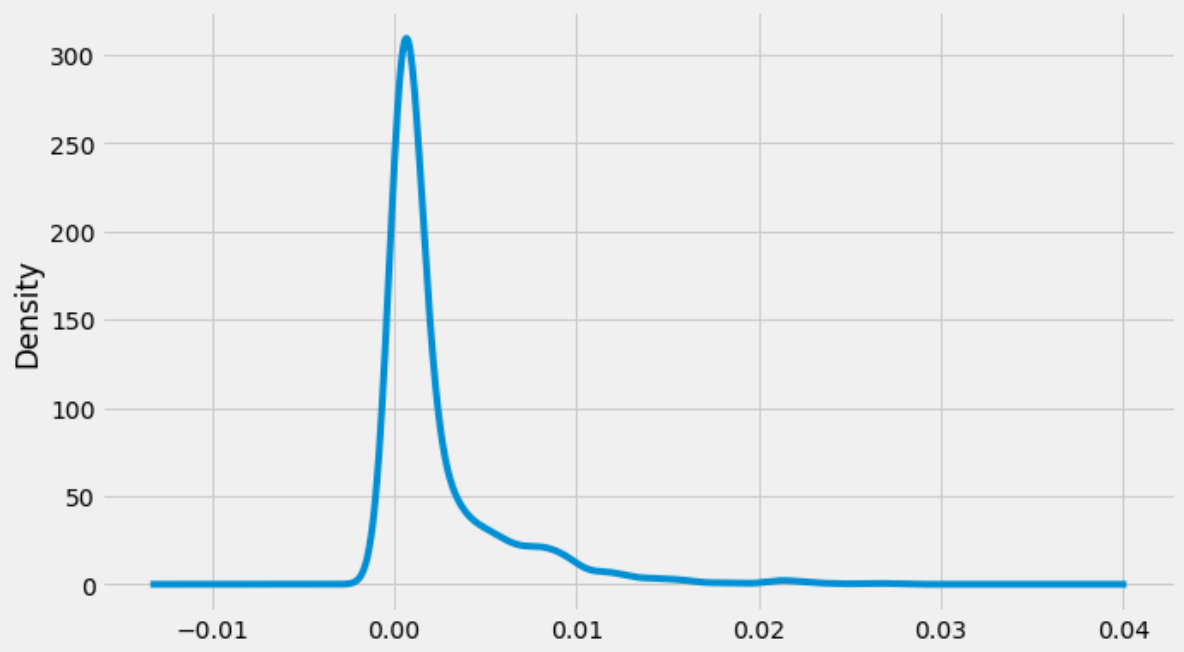


In [76]:

```
rolstd.plot(kind='kde')
```

Out[76]:

```
<AxesSubplot:ylabel='Density'>
```



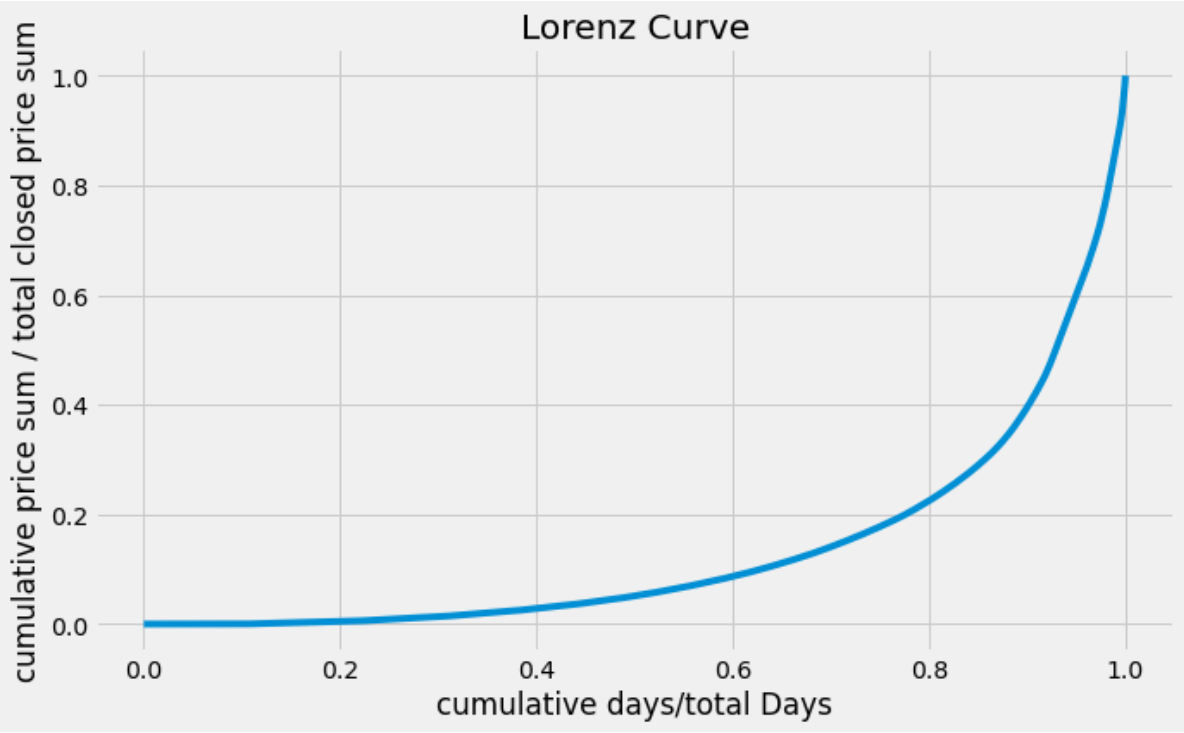
In [82]:

```
Closed_price= df['Price']
Close_price=[];
Closed_diff =[]
closed_diff_p1=[]
kval=[]
kval2=[]

'''

difference between closed prices of two days

'''
for i in range((len(Closed_price)-1)):
    j=abs(Closed_price[i+1] - Closed_price[i])
    Closed_diff.append(j)
    closed_diff_p1.append(j/Closed_price[i]);

j =0
for i in Closed_price:
    Close_price.append(i)
for i in range(0,len(Close_price),1 ):
        lis= Closed_diff[:i]
        lis2 = closed_diff_p1[:i]
        kval.append(lis)
        kval2.append(lis2)
        j =i
if(len(Close_price) > j):
    kval.append( Closed_diff)
    kval2.append(closed_diff_p1)




'''

window preparation

'''
window=[]
window2=[]
for i in range(0,len(Close_price)-20):
        lis=Closed_diff[i:i+20]
        lis2= closed_diff_p1[i:i+20]
        window2.append(lis2)
        window.append(lis)
        j =i



'''

lorenz curve

'''
```

```
sum1 =0;
total_sum =0;
y_axis=[]
for i in Closed_diff:
    total_sum+= i;



Closed_diff.sort()
y_axis=[]


n = len(Closed_diff)
x_axis=[]
count=0
for  i in Closed_diff:
    sum1+= i
    count+=1
    y_axis.append(sum1/total_sum);
    x_axis.append(count/n);
plt.title("Lorenz Curve");
plt.xlabel('cumulative days/total Days')
plt.ylabel('cumulative price sum / total closed price sum')
plt.plot(x_axis,y_axis)
```

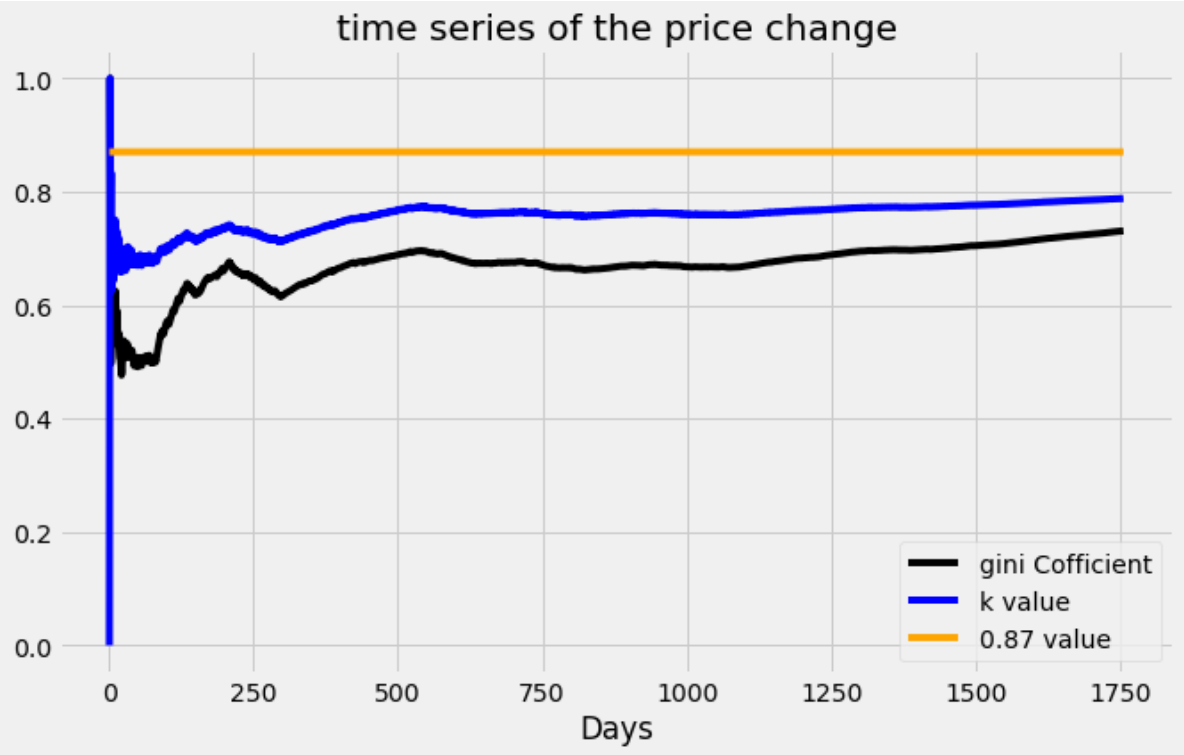
Out[82]:

```
[<matplotlib.lines.Line2D at 0x17bbc2b8d00>]
```



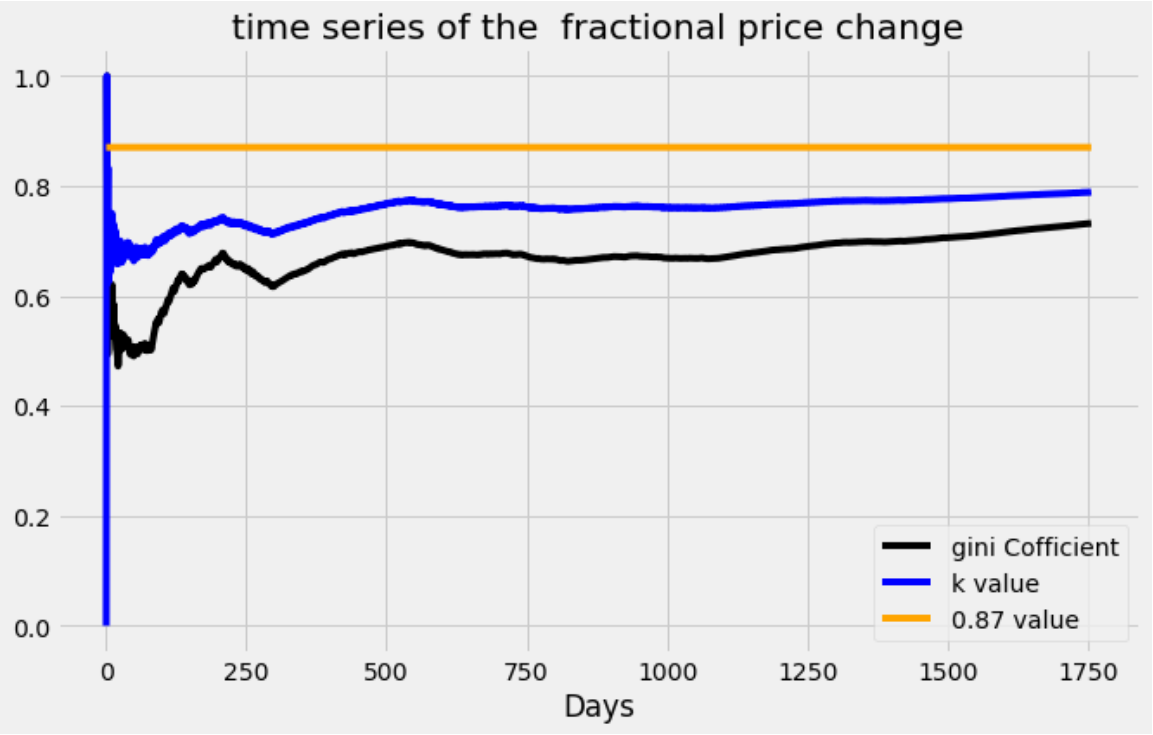
In [83]:

```
ginilis=[]
kvlist=[]
lenval=[]
val87=[]
count=0
def lorenz(x):
    y_axis=[]
    x.sort()
    n = len(x)
    x_axis=[]
    count=0
    sum1 =0
    total_sum =0;
    for i in x:
        total_sum+= i;
    for  i in x:
        sum1+= i
        count+=1
        y_axis.append(sum1/total_sum);
        x_axis.append(count/n);
    return x_axis,y_axis
def kv(x):
    diff=[]
    value=0
    x_axis1 , y_axis1 = lorenz(x);
    for i in range(len(x)):
        d=(1-x_axis1[i]) - y_axis1[i]
        if(d < 0 ):
            value= (x_axis1[i-1] + x_axis1[i])/2
            break;
        diff.append(d)
    return value;

def gini(x):
    sum2 =0
    x_axis1 , y_axis1 = lorenz(x);
    for i in range(len(x)-1):
        sum2 = sum2+(( y_axis1[i+1]+  y_axis1[i] )/2)*(x_axis1[i+1]-x_axis1[i]))
    ginival =1- (2 * sum2)
    return ginival
for i in kval:
    count +=1
    ginilis.append(gini(i))
    kvlist.append(kv(i))
    lenval.append(count)
    val87.append(0.87)
plt.xlabel('Days')
plt.plot(lenval,ginilis,color="Black",label="gini Cofficient")
plt.plot(lenval,kvlist,color="Blue",label="k value")
plt.plot(lenval,val87,color="orange",label="0.87 value")
plt.legend(loc='best')
plt.title("time series of the price change")
plt.show(block=False)
```


time series of the price change
1.0
0.8
0.6
0.4
0.2
0.0
0
250
500
750
1000
1250
1500
1750
Days
gini Cofficient
k value
0.87 value

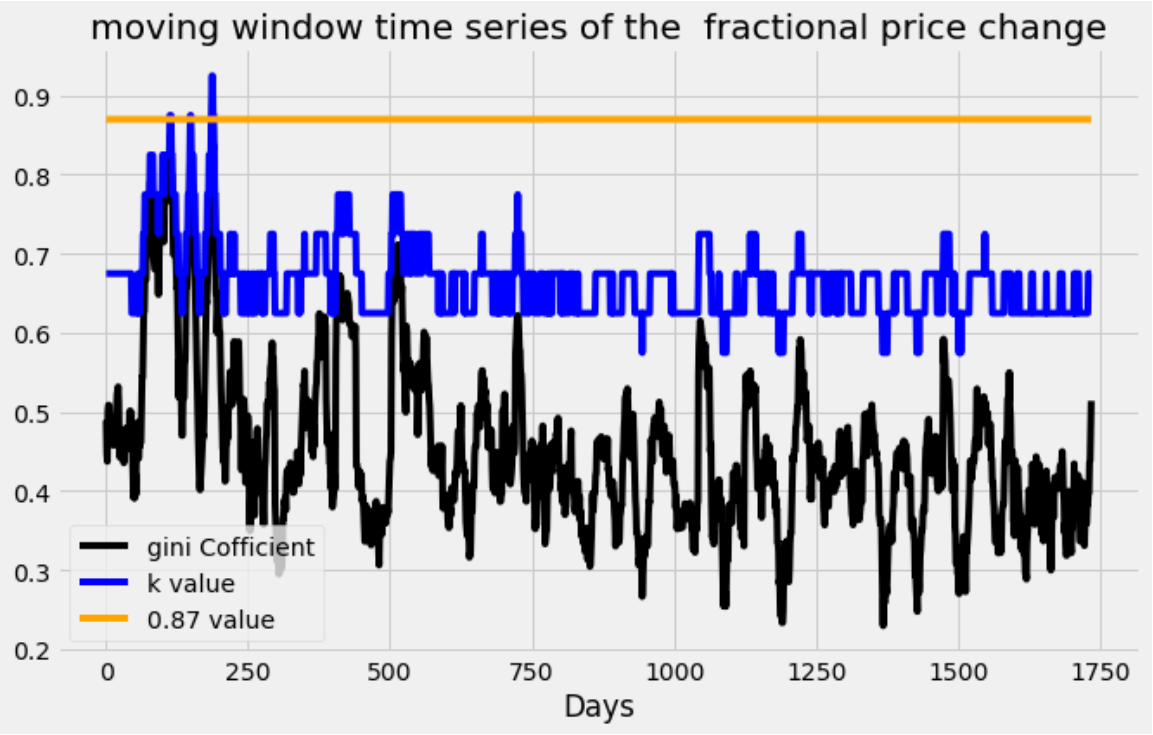
In [84]:

```
ginilis=[]
kvlist=[]
lenval=[]
val87=[]
count=0
def lorenz(x):
    y_axis=[]
    x.sort()
    n = len(x)
    x_axis=[]
    count=0
    sum1 =0
    total_sum =0;
    for i in x:
        total_sum+= i;
    for  i in x:
        sum1+= i
        count+=1
        y_axis.append(sum1/total_sum);
        x_axis.append(count/n);
    return x_axis,y_axis
def kv(x):
    diff=[]
    value=0
    x_axis1 , y_axis1 = lorenz(x);
    for i in range(len(x)):
        d=(1-x_axis1[i]) - y_axis1[i]
        if(d < 0 ):
            value= (x_axis1[i-1] + x_axis1[i])/2
            break;
        diff.append(d)
    return value;

def gini(x):
    sum2 =0
    x_axis1 , y_axis1 = lorenz(x);
    for i in range(len(x)-1):
        sum2 = sum2+((( y_axis1[i+1]+  y_axis1[i] )/2)*(x_axis1[i+1]-x_axis1[i]))
    ginival =1- (2 * sum2)
    return ginival
for i in kval2:
    count +=1
    ginilis.append(gini(i))
    kvlist.append(kv(i))
    lenval.append(count)
    val87.append(0.87)
plt.xlabel('Days')
plt.plot(lenval,ginilis,color="Black",label="gini Cofficient")
plt.plot(lenval,kvlist,color="Blue",label="k value")
plt.plot(lenval,val87,color="orange",label="0.87 value")
plt.legend(loc='best')
plt.title("time series of the  fractional price change")
plt.show(block=False)
```


time series of the fractional price change
1.0
0.8
0.6
0.4
0.2
0.0
0
250
500
750
1000
1250
1500
1750
Days
gini Cofficient
k value
0.87 value

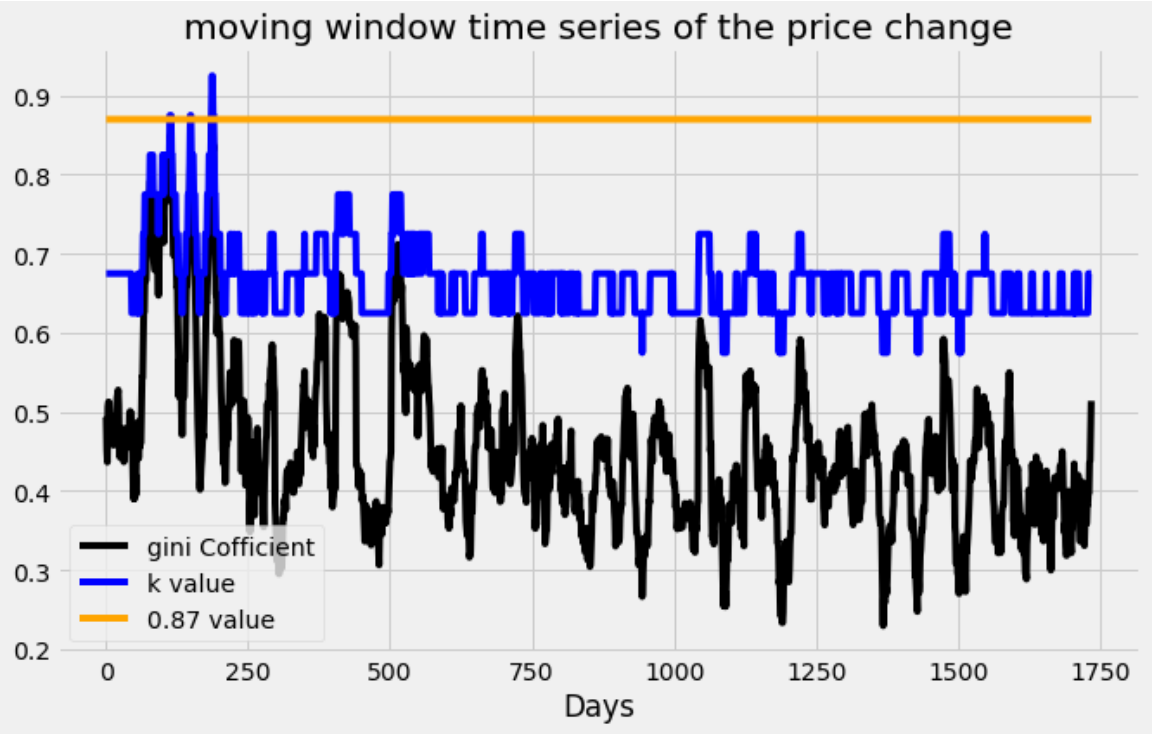
In [85]:

```
ginilis=[]
kvlist=[]
lenval=[]
val87=[]
count=0
def lorenz(x):
    y_axis=[]
    x.sort()
    n = len(x)
    x_axis=[]
    count=0
    sum1 =0
    total_sum =0;
    for i in x:
        total_sum+= i;
    for  i in x:
        sum1+= i
        count+=1
        y_axis.append(sum1/total_sum);
        x_axis.append(count/n);
    return x_axis,y_axis
def kv(x):
    diff=[]
    value=0
    x_axis1 , y_axis1 = lorenz(x);
    for i in range(len(x)):
        d=(1-x_axis1[i]) - y_axis1[i]
        if(d < 0 ):
            value= (x_axis1[i-1] + x_axis1[i])/2
            break;
        diff.append(d)
    return value;

def gini(x):
    sum2 =0
    x_axis1 , y_axis1 = lorenz(x);
    for i in range(len(x)-1):
        sum2 = sum2+((( y_axis1[i+1]+  y_axis1[i] )/2)*(x_axis1[i+1]-x_axis1[i]))
    ginival =1- (2 * sum2)
    return ginival
for i in window2:
    count +=1
    ginilis.append(gini(i))
    kvlist.append(kv(i))
    lenval.append(count)
    val87.append(0.87)


plt.xlabel('Days')
plt.plot(lenval,ginilis,color="Black",label="gini Cofficient")
plt.plot(lenval,kvlist,color="Blue",label="k value")
plt.plot(lenval,val87,color="orange",label="0.87 value")
plt.legend(loc='best')
plt.title("moving window time series of the  fractional price change")
plt.show(block=False)
```


moving window time series of the fractional price change
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
gini Cofficient
k value
0.87 value
0
250
500
750
1000
1250
1500
1750
Days

In [86]:

```
ginilis=[]
kvlist=[]
lenval=[]
val87=[]
count=0
def lorenz(x):
    y_axis=[]
    x.sort()
    n = len(x)
    x_axis=[]
    count=0
    sum1 =0
    total_sum =0;
    for i in x:
        total_sum+= i;
    for  i in x:
        sum1+= i
        count+=1
        y_axis.append(sum1/total_sum);
        x_axis.append(count/n);
    return x_axis,y_axis
def kv(x):
    diff=[]
    value=0
    x_axis1 , y_axis1 = lorenz(x);
    for i in range(len(x)):
        d=(1-x_axis1[i]) - y_axis1[i]
        if(d < 0 ):
            value= (x_axis1[i-1] + x_axis1[i])/2
            break;
        diff.append(d)
    return value;

def gini(x):
    sum2 =0
    x_axis1 , y_axis1 = lorenz(x);
    for i in range(len(x)-1):
        sum2 = sum2+((( y_axis1[i+1]+  y_axis1[i] )/2)*(x_axis1[i+1]-x_axis1[i]))
    ginival =1- (2 * sum2)
    return ginival
for i in window:
    count +=1
    ginilis.append(gini(i))
    kvlist.append(kv(i))
    lenval.append(count)
    val87.append(0.87)


plt.xlabel('Days')
plt.plot(lenval,ginilis,color="Black",label="gini Cofficient")
plt.plot(lenval,kvlist,color="Blue",label="k value")
plt.plot(lenval,val87,color="orange",label="0.87 value")
plt.legend(loc='best')
plt.title("moving window time series of the price change")
plt.show(block=False)
```


moving window time series of the price change
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
gini Cofficient
k value
0.87 value
0
250
500
750
1000
1250
1500
1750
Days

In [ ]:

In [ ]:

In [ ]:

In [ ]: