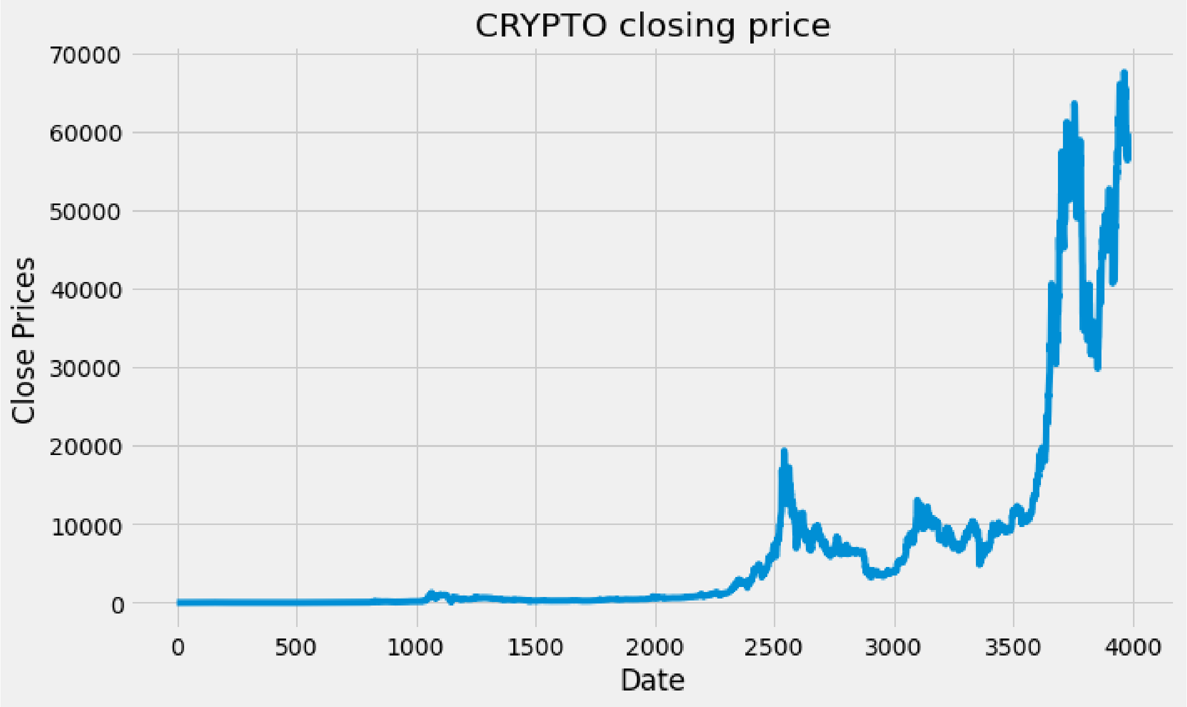
In [23]:

```
"""
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("Bitcoin.csv");
print(df)
```

```
              Date    Price      Open      High       Low    Vol. Change %  
\
0     Jan 01, 2011      0.3       0.3       0.3       0.3   2.82K    0.00%   
1     Jan 02, 2011      0.3       0.3       0.3       0.3   5.35K    0.00%   
2     Jan 03, 2011      0.3       0.3       0.3       0.3   1.43K    0.00%   
3     Jan 04, 2011      0.3       0.3       0.3       0.3   1.88K    0.00%   
4     Jan 05, 2011      0.3       0.3       0.3       0.3   0.36K    0.00%   
...            ...      ...       ...       ...       ...     ...      ...   
3976  Nov 20, 2021  59717.6  58,080.8  59,854.6  57,435.3  44.53K    2.81%   
3977  Nov 21, 2021  58714.3  59,724.0  60,002.6  58,528.3  40.98K   -1.68%   
3978  Nov 22, 2021  56308.8  58,719.7  59,288.3  55,656.8  76.45K   -4.10%   
3979  Nov 23, 2021  57573.2  56,304.8  57,855.1  55,542.5  72.48K    2.25%   
3980  Nov 24, 2021  56517.8  57,559.4  57,697.4  56,171.3  68.16K   -1.83%   

          Gini    Kvalue  
0     1.000000  0.000000  
1     1.000000  0.000000  
2          NaN  0.000000  
3          NaN  0.000000  
4          NaN  0.000000  
...        ...       ...  
3976  0.844233  0.844190  
3977  0.844117  0.843978  
3978  0.844164  0.844268  
3979  0.844068  0.844056  
3980  0.843955  0.844095  

[3981 rows x 9 columns]
```

In [26]:

```
'''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()
```

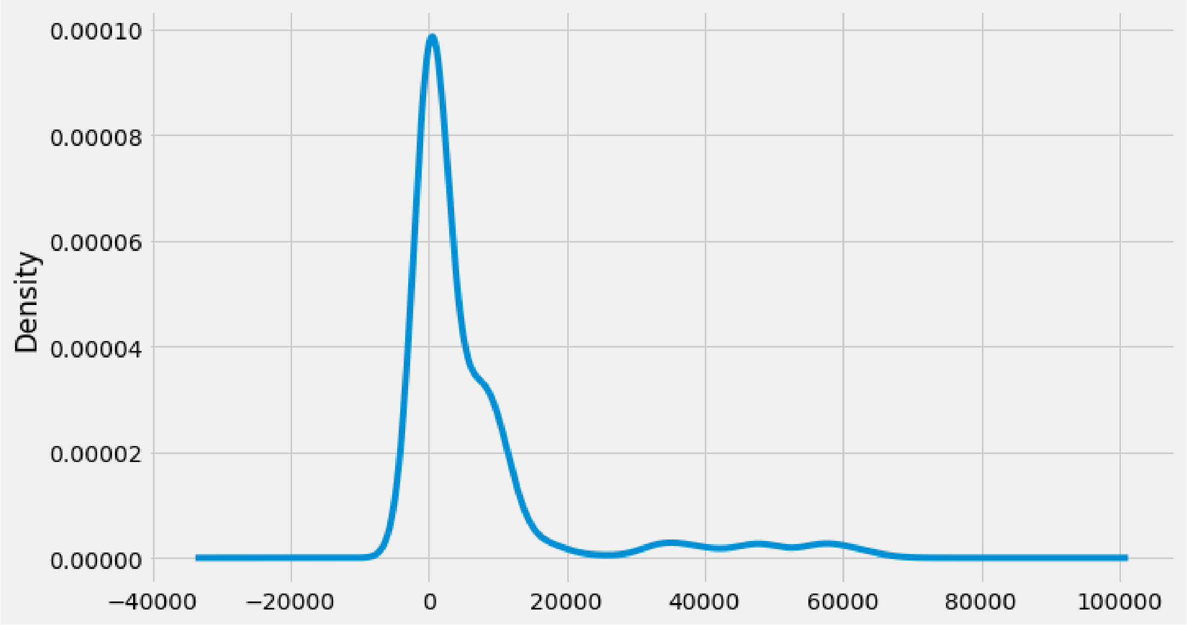


In [27]:

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

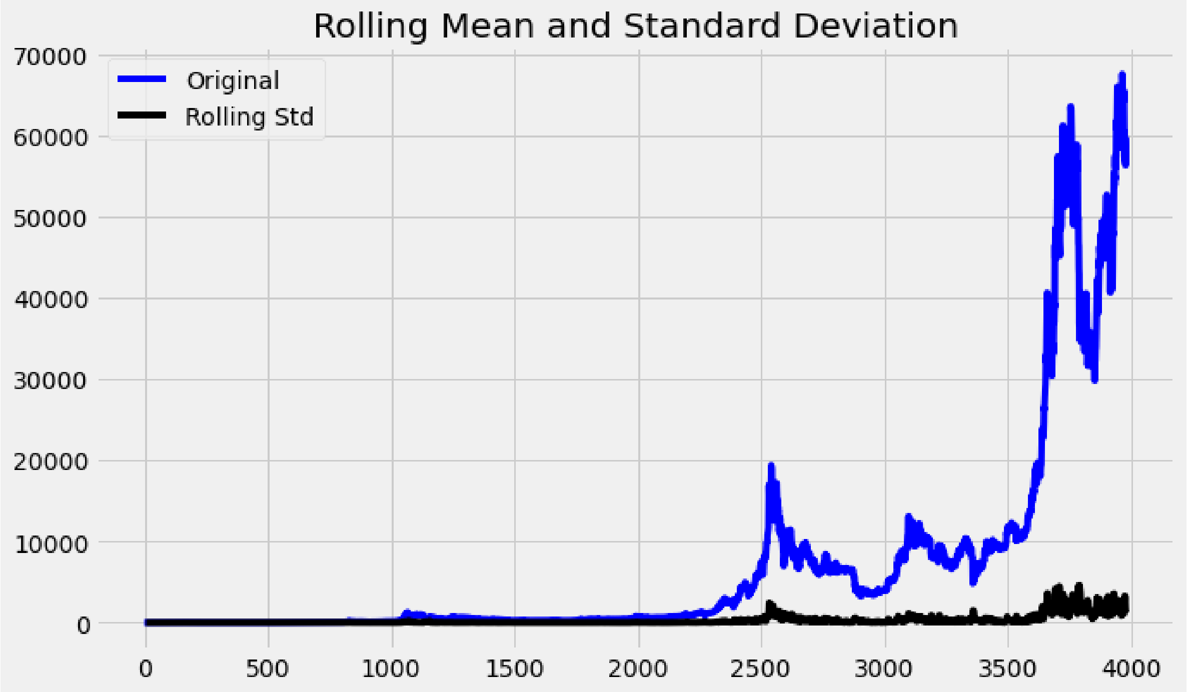
Out[27]:

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



In [28]:

```
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)
```

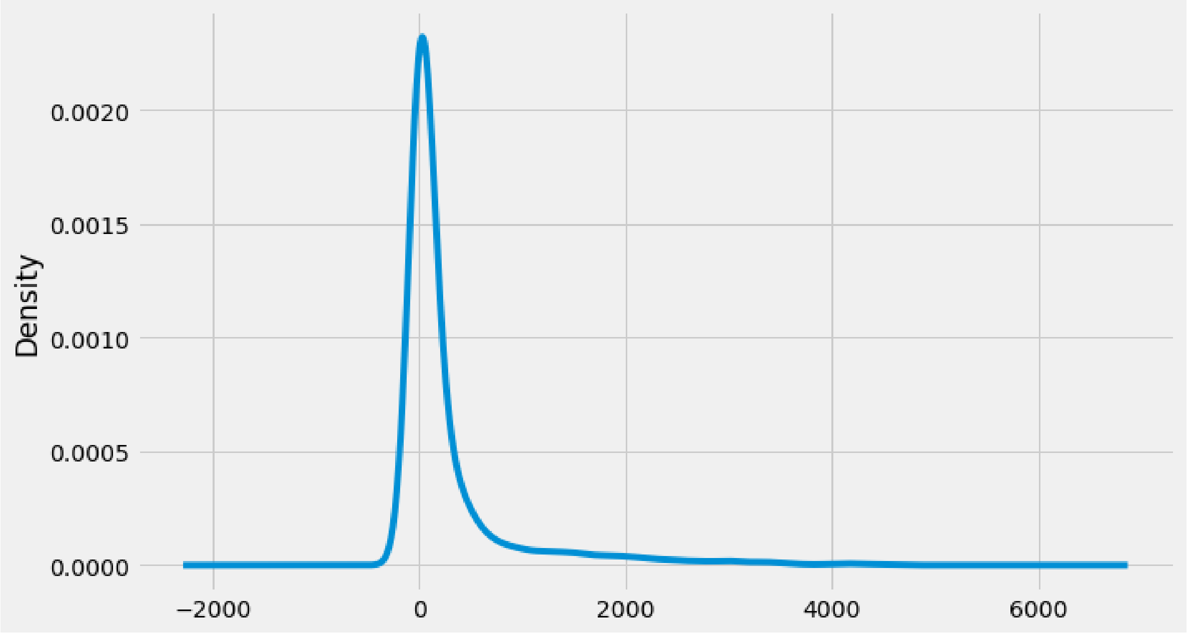


In [29]:

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

Out[29]:

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



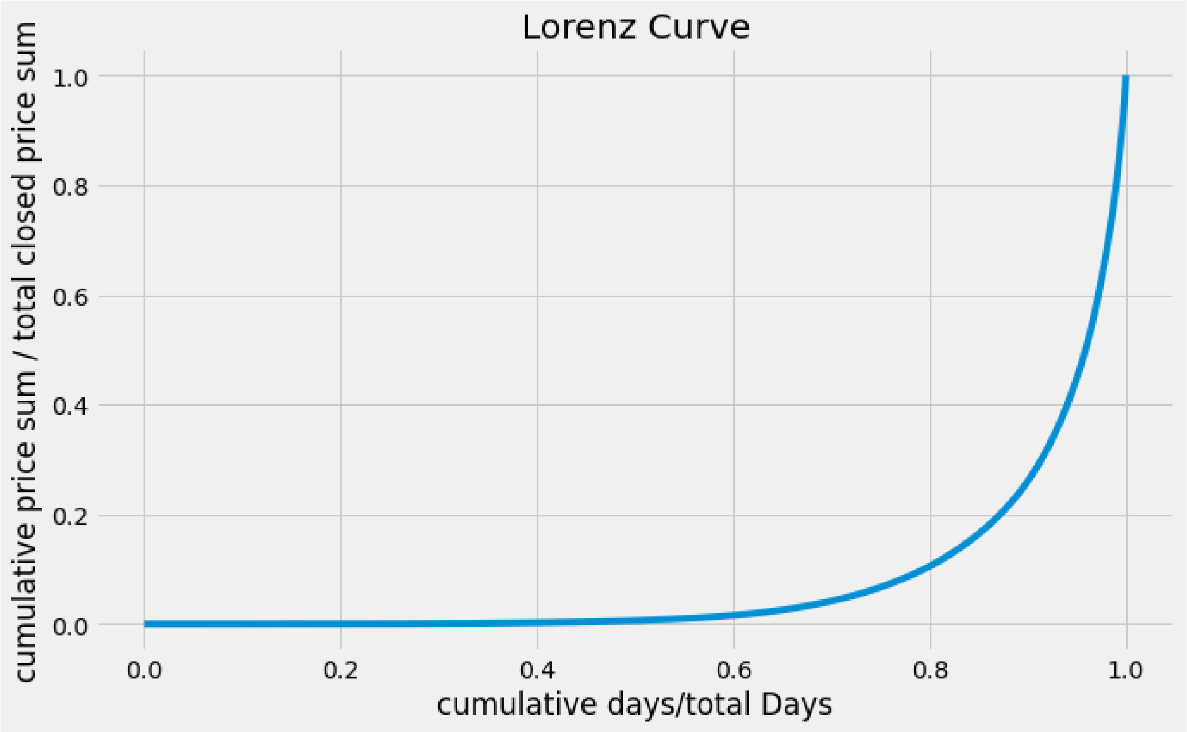
In [41]:

```
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)
```

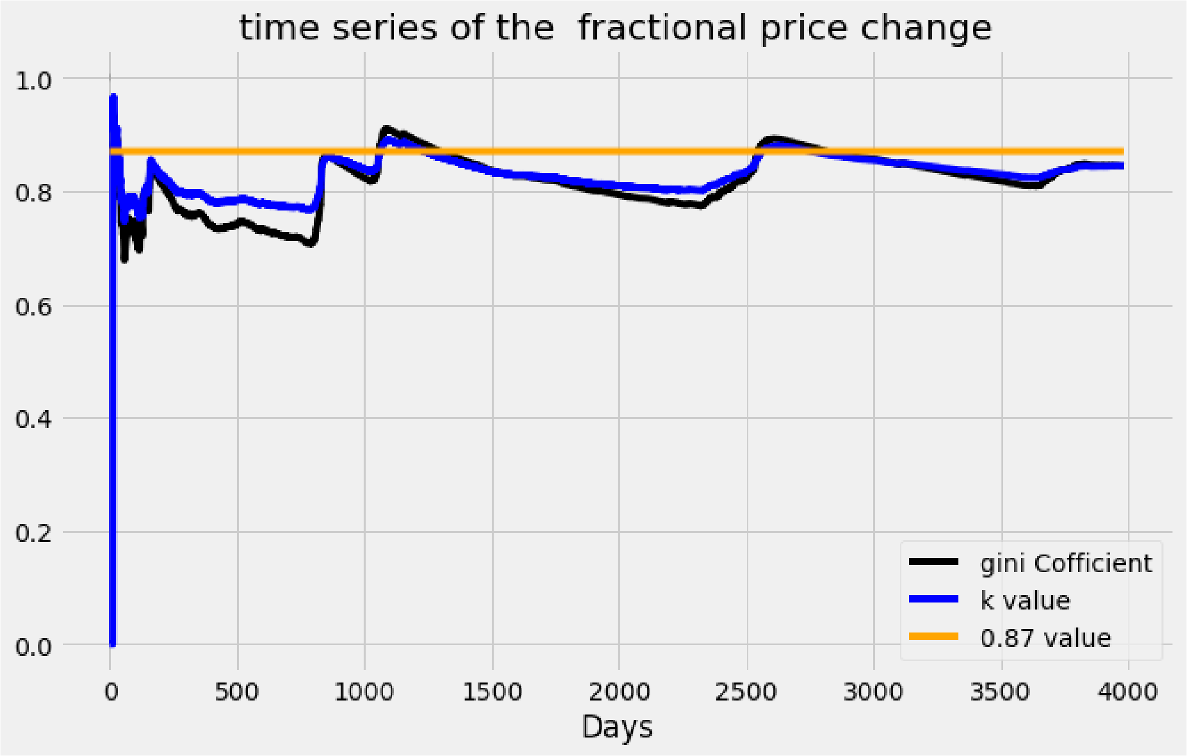
Out[41]:

```
[<matplotlib.lines.Line2D at 0x17bbe24e850>]
```



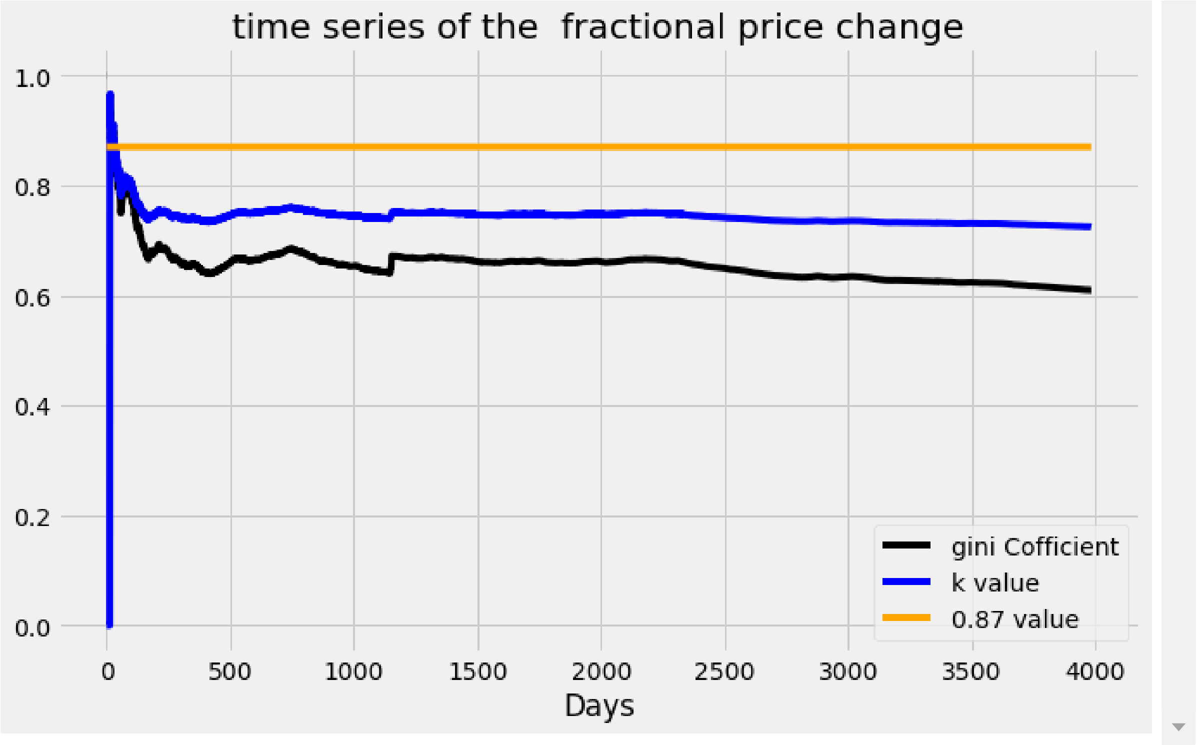
In [42]:

```
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 fractional price change
1.0
0.8
0.6
0.4
0.2
0.0
0
500
1000
1500
2000
2500
3000
3500
4000
Days
gini Cofficient
k value
0.87 value

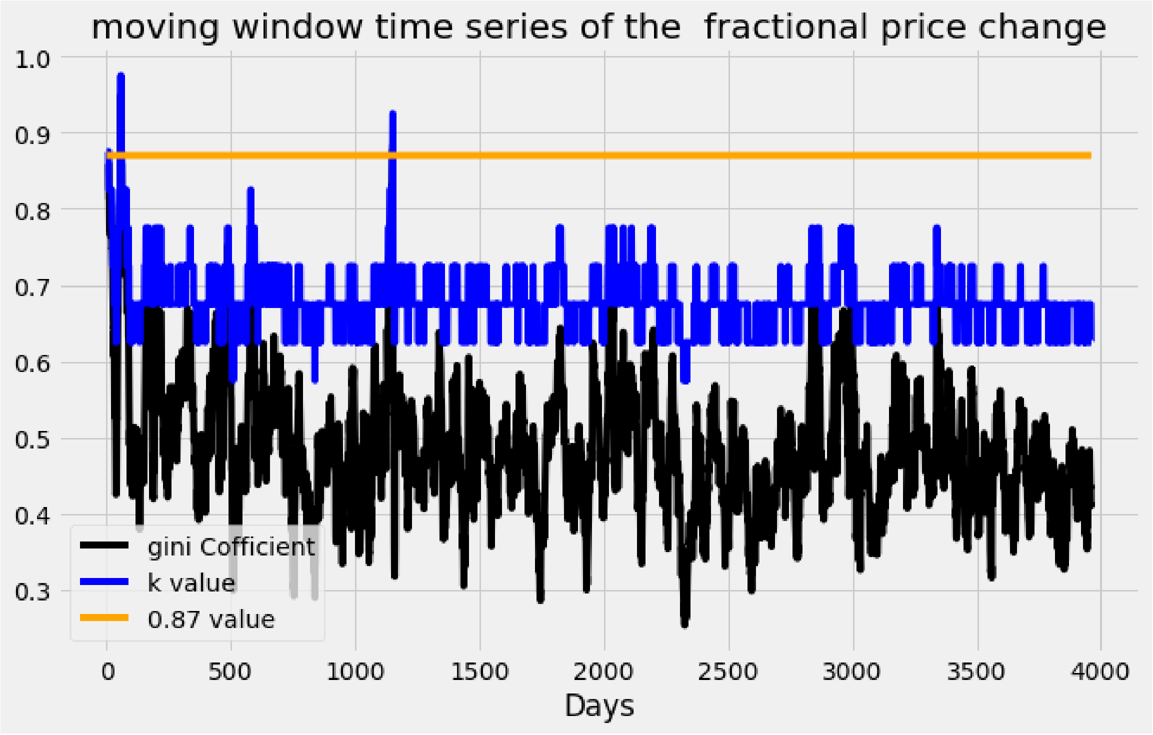
In [43]:

```
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
500
1000
1500
2000
2500
3000
3500
4000
Days
gini Cofficient
k value
0.87 value

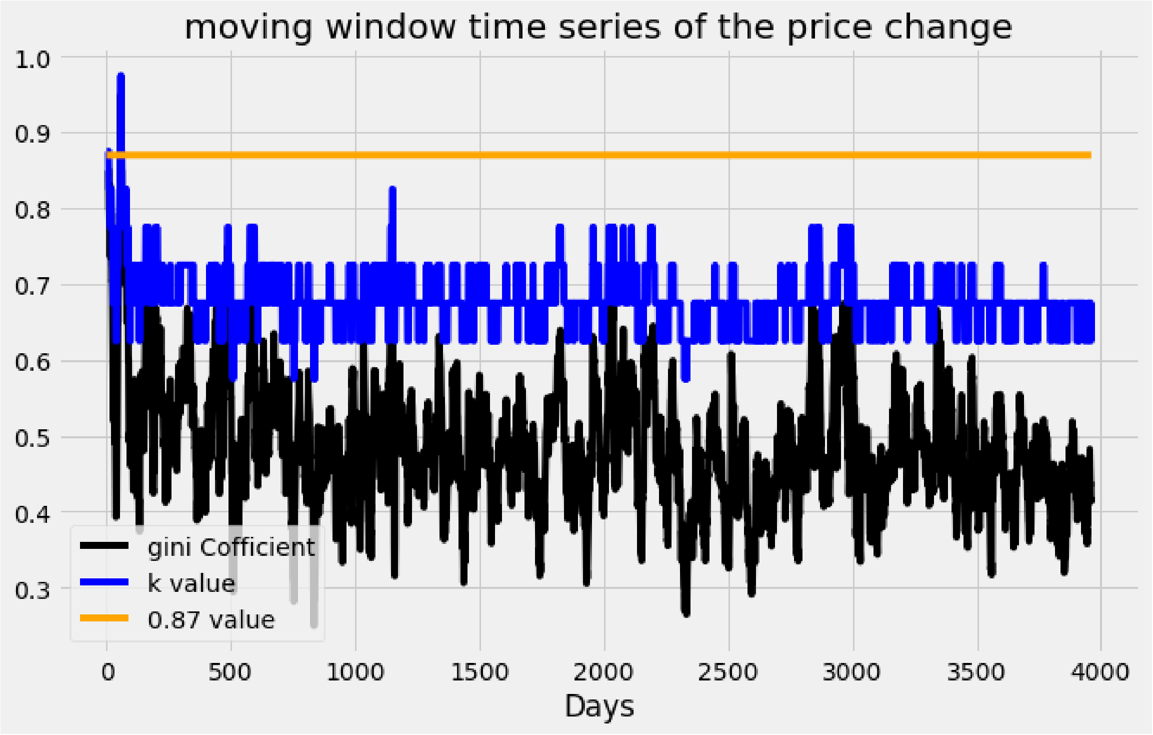
In [32]:

```
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
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
500
1000
1500
2000
2500
3000
3500
4000
Days
gini Cofficient
k value
0.87 value

In [33]:

```
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
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
gini Cofficient
k value
0.87 value
0
500
1000
1500
2000
2500
3000
3500
4000
Days

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```