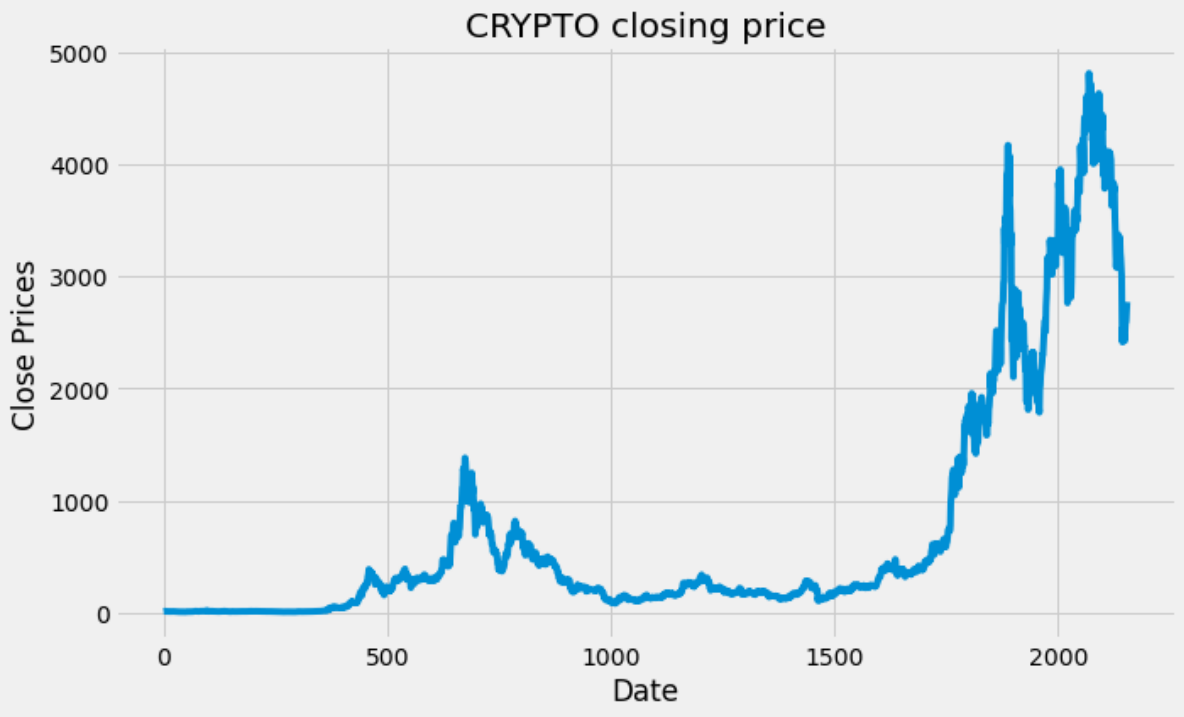
In [57]:

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

```
             Date     Price      Open      High       Low     Vol. Change %
0     Mar 10, 2016     11.75     11.20     11.85     11.07    0.00K    4.91%
1     Mar 11, 2016     11.95     11.75     11.95     11.75    0.18K    1.70%
2     Mar 12, 2016     12.92     11.95     13.45     11.95    0.83K    8.12%
3     Mar 13, 2016     15.07     12.92     15.07     12.92    1.30K   16.64%
4     Mar 14, 2016     12.50     15.07     15.07     11.40   92.18K  -17.05%
...            ...       ...       ...       ...       ...      ...      ...
2150  Jan 28, 2022  2,545.95  2,424.66  2,556.34  2,362.53  634.50K    5.00%
2151  Jan 29, 2022  2,600.02  2,545.95  2,632.51  2,520.14  484.18K    2.12%
2152  Jan 30, 2022  2,602.77  2,600.12  2,636.04  2,543.99  381.84K    0.11%
2153  Jan 31, 2022  2,686.82  2,602.71  2,704.64  2,478.85  601.73K    3.23%
2154  Feb 01, 2022  2,769.14  2,686.82  2,807.35  2,676.32  623.55K    3.06%

[2155 rows x 7 columns]
```

In [59]:

```
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()
```

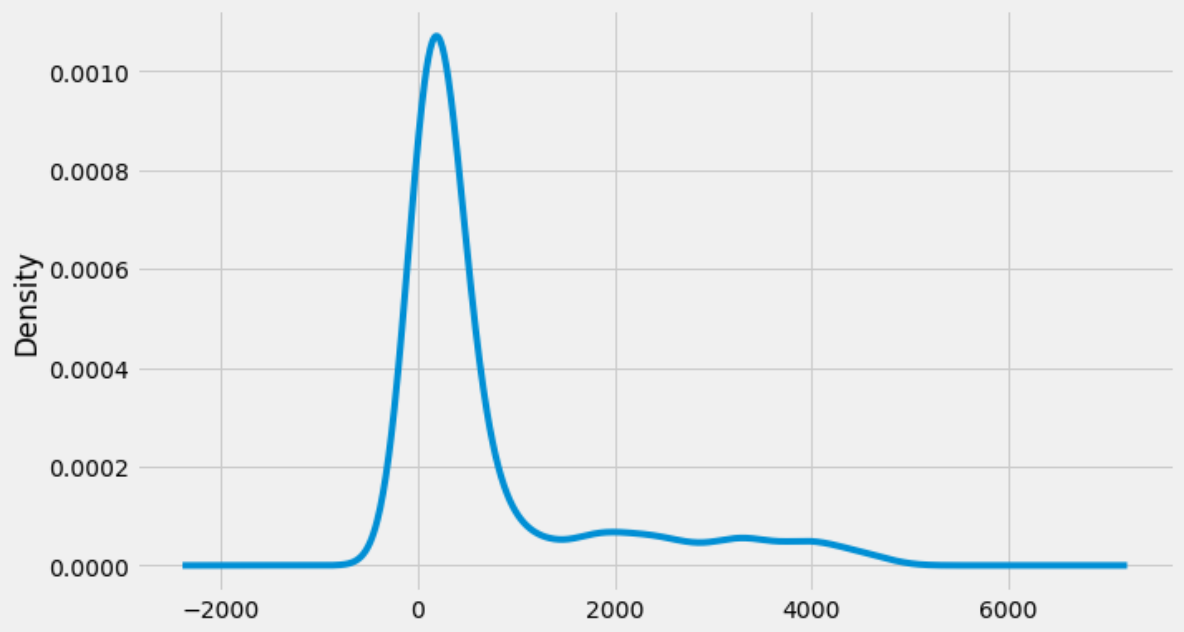


In [60]:

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

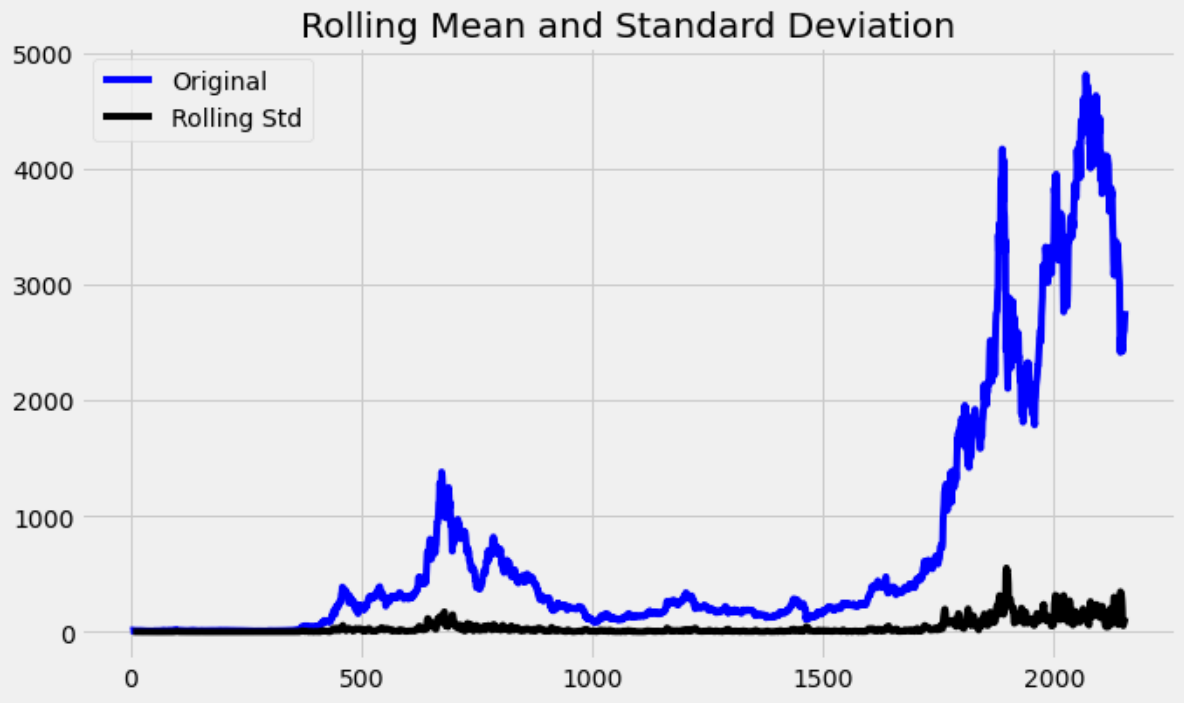
Out[60]:

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



In [61]:

```
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)
```

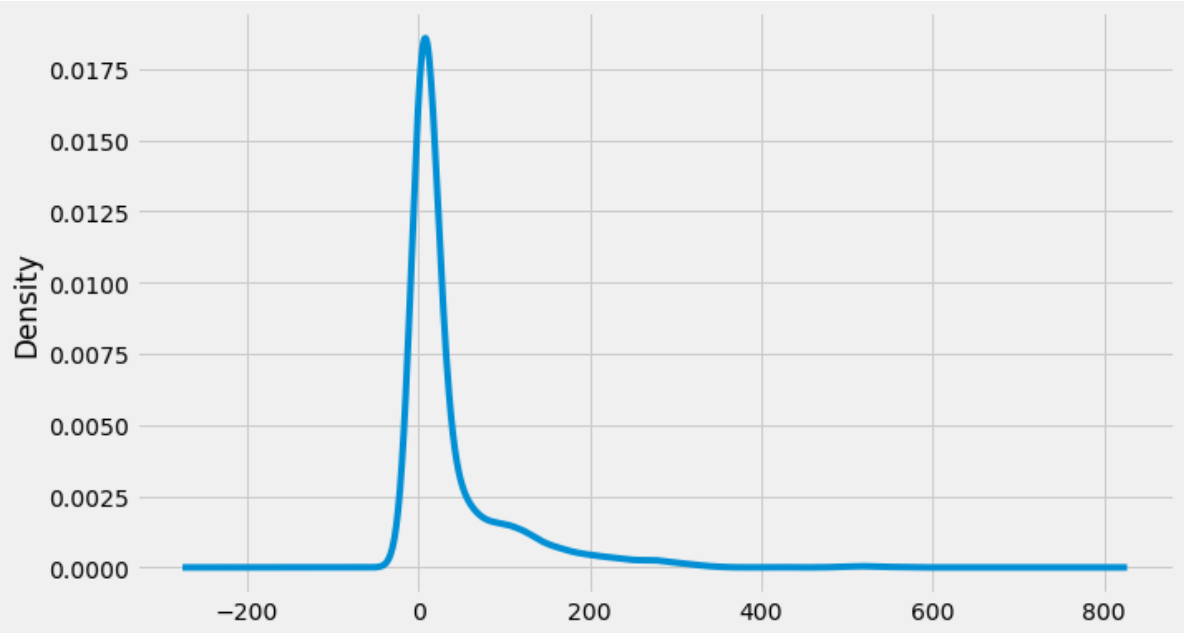


In [62]:

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

Out[62]:

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



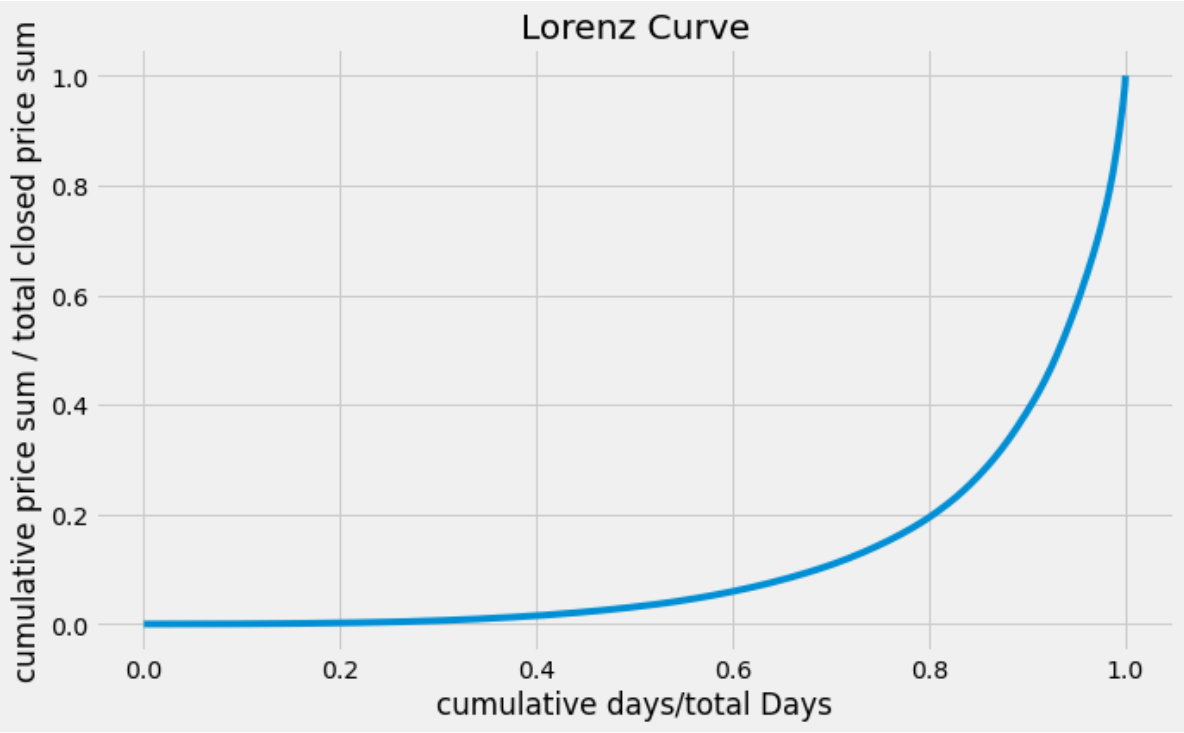
In [63]:

```
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)
```

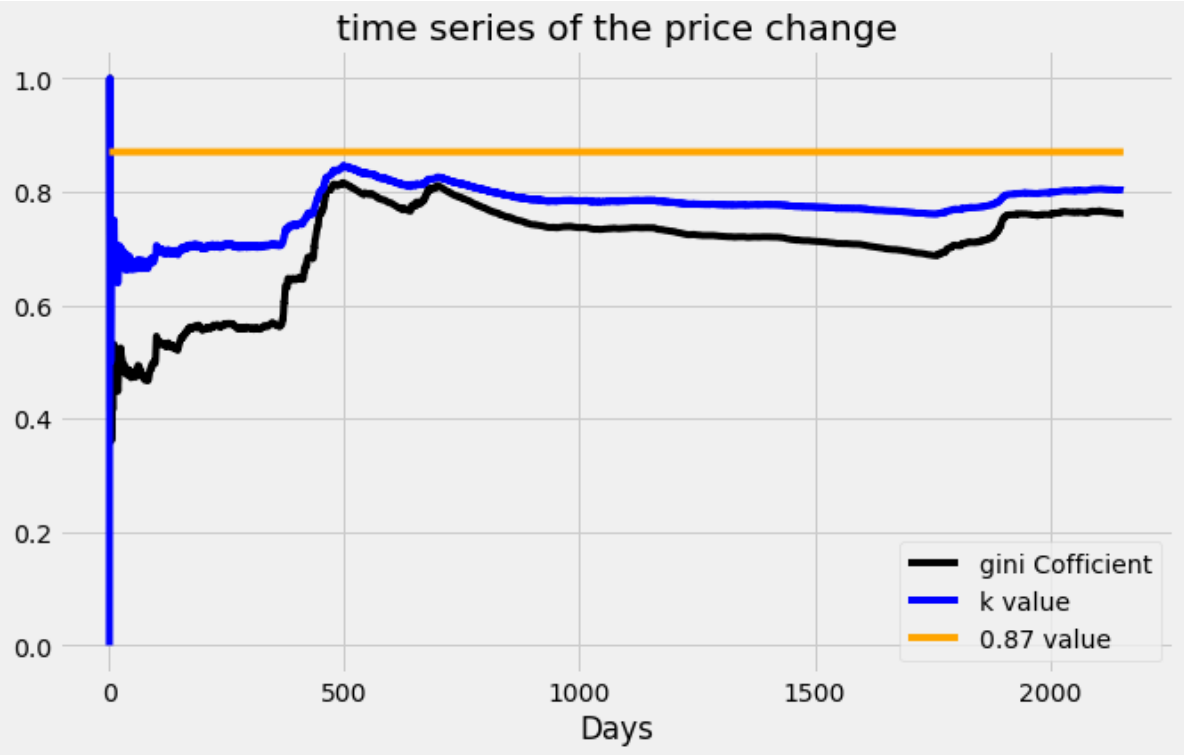
Out[63]:

```
[<matplotlib.lines.Line2D at 0x17bbdc529a0>]
```



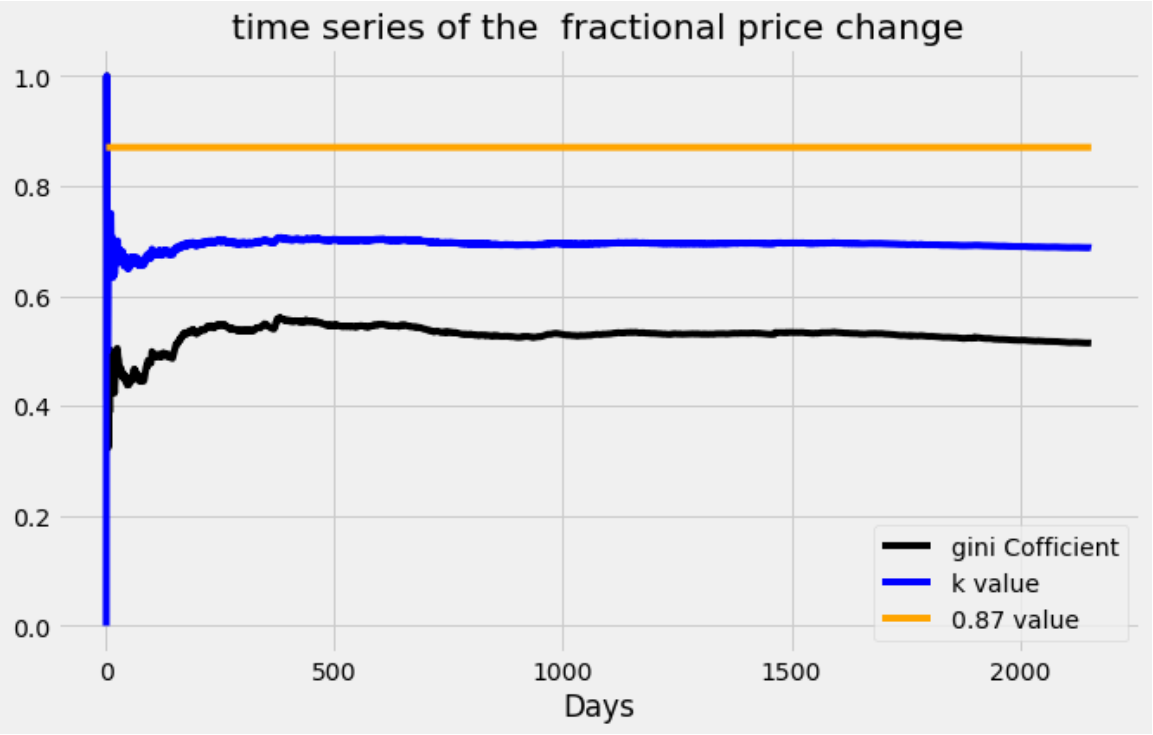
In [64]:

```
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
500
1000
1500
2000
Days
gini Cofficient
k value
0.87 value

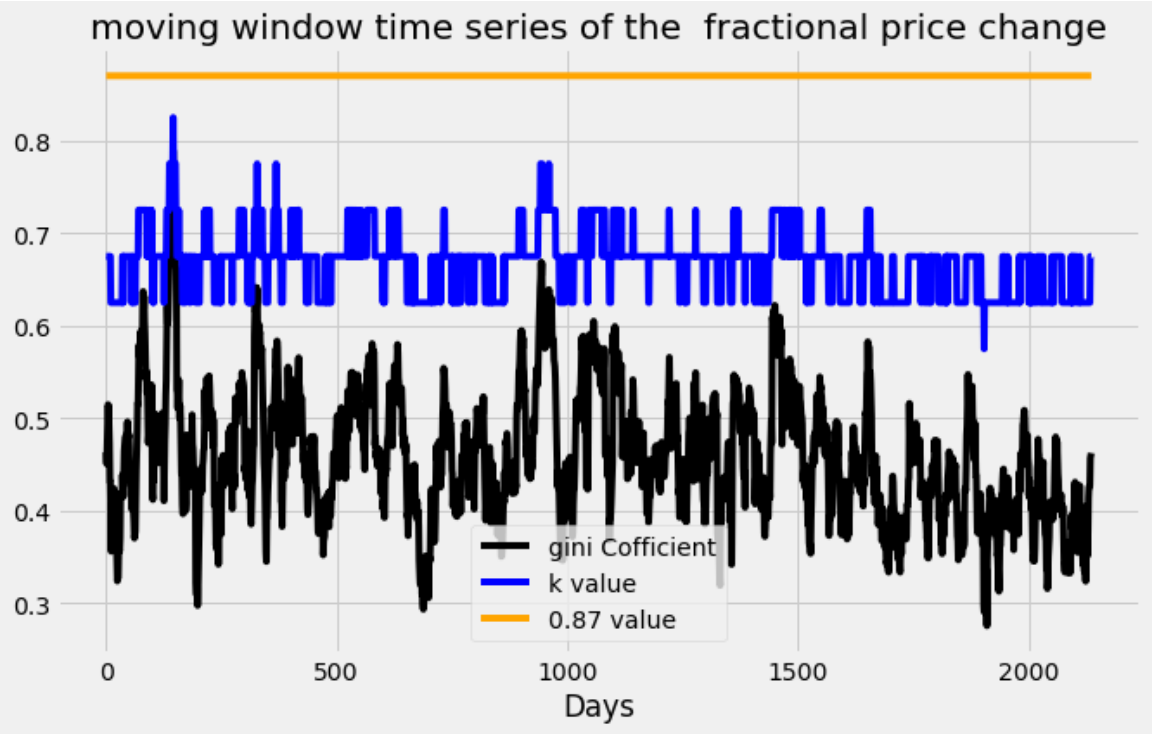
In [67]:

```
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
Days
gini Cofficient
k value
0.87 value

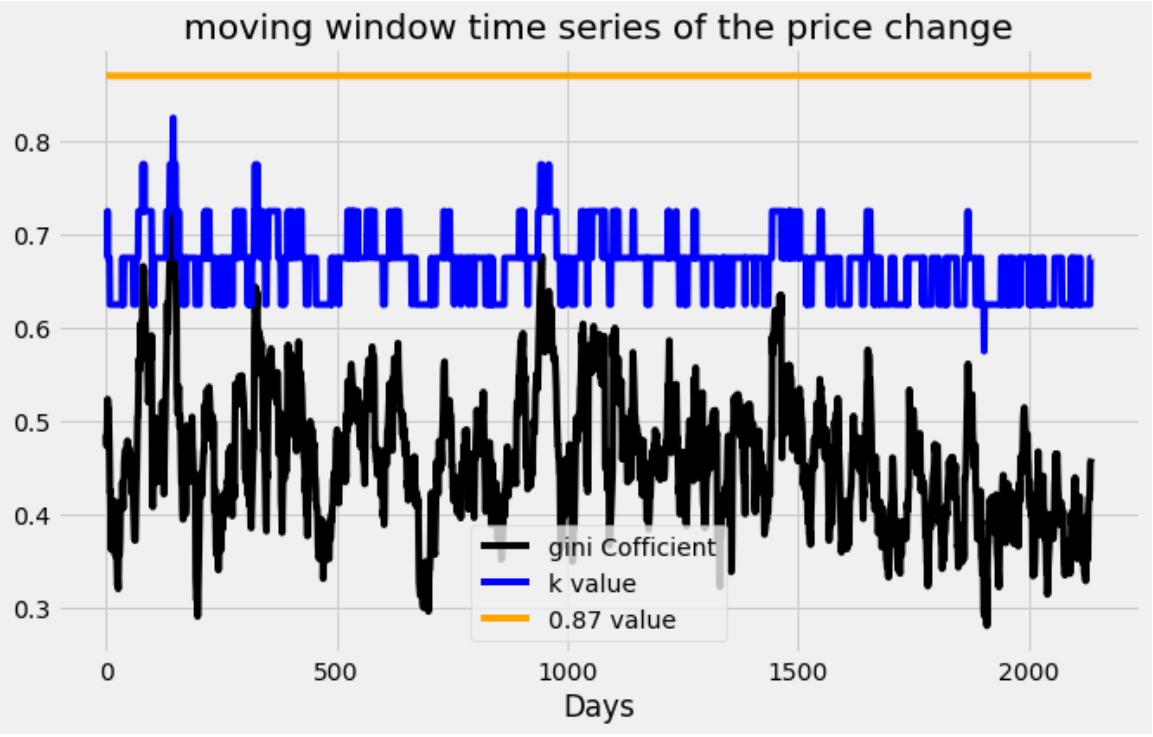
In [68]:

```
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
0.8
0.7
0.6
0.5
0.4
0.3
0
500
1000
1500
2000
Days
gini Cofficient
k value
0.87 value

In [69]:

```
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
0.8
0.7
0.6
0.5
0.4
0.3
0
500
1000
1500
2000
Days
gini Cofficient
k value
0.87 value

In [ ]:

In [ ]:

In [ ]:

In [ ]: