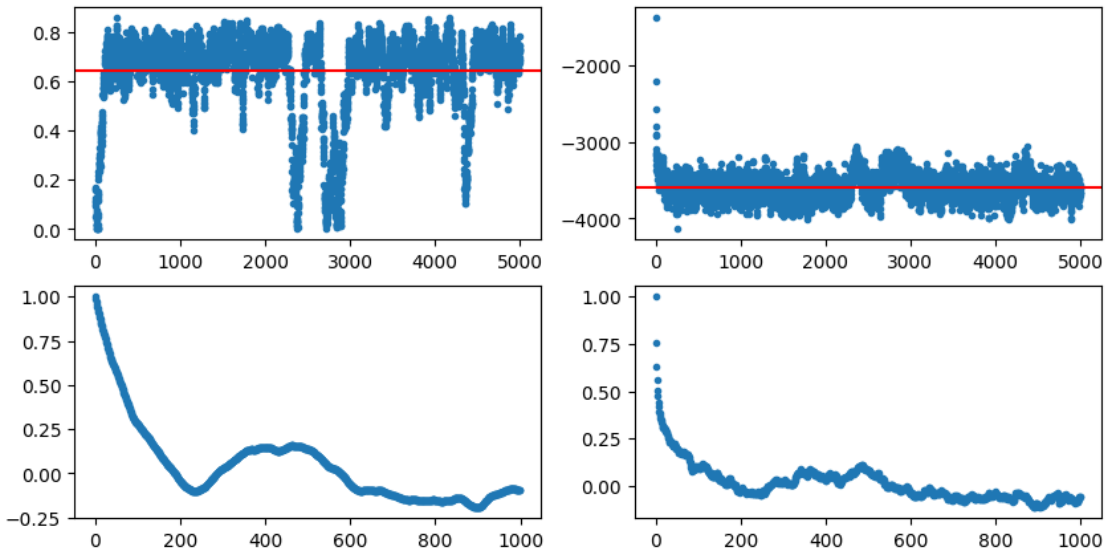
analysis

December 20, 2023

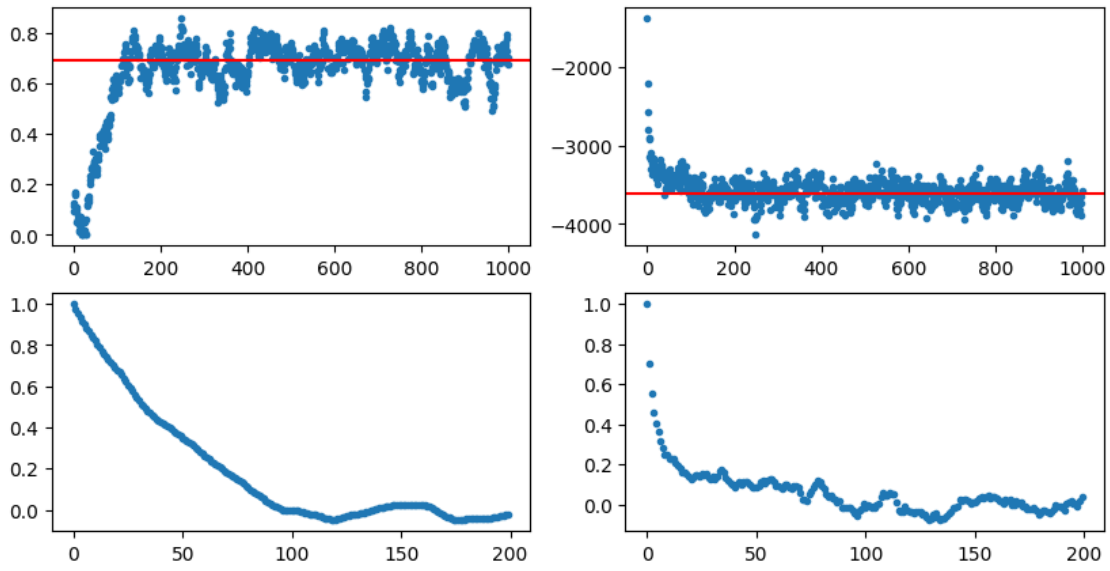
[1]:
```
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.graphics.tsaplots import plot_acf

def data_blocking(data,block_length):
    N = data.size
    l = N//block_length
    blocked_data = np.empty(l)

    for i in range(l):
        x_i = 0
        for j in range(block_length):
            x_i += data[(i-1)*block_length + j]

        blocked_data[i]= x_i/block_length


    return blocked_data

def bootstrap_data(data):
    N = data.size
    new_data = np.empty(N)
    for i in range(N):
        new_data[i] = data[np.random.randint(0,N-1)]
    return new_data

def autocorr(data,lag):
    mean = np.mean(data)
    C_0 = np.mean((data-mean)**2)
    if(lag>0):
        C_X = np.mean((data[lag:]-mean)*(data[:-lag]-mean))
    else:
        C_X = np.mean((data-mean)**2)
    return C_X/C_0

def autocorr_array(data,lag):
    x= np.zeros(lag)
```

```
    for i in range(lag):
        x[i] = autocorr(data,i)
    return x

def data_blocking_error(data,length):
    result = np.zeros((3,length))
    for i in range(1,length):
        N =len(data)
        x = []
        for j in range(0,N,i):
            a = data[j:j+i]
            x.append(np.std(a))
        result[0,i] = i
        result[1,i] = np.mean(x)
        result[2,i] = np.std(x)/np.sqrt(len(x))

    return result
```

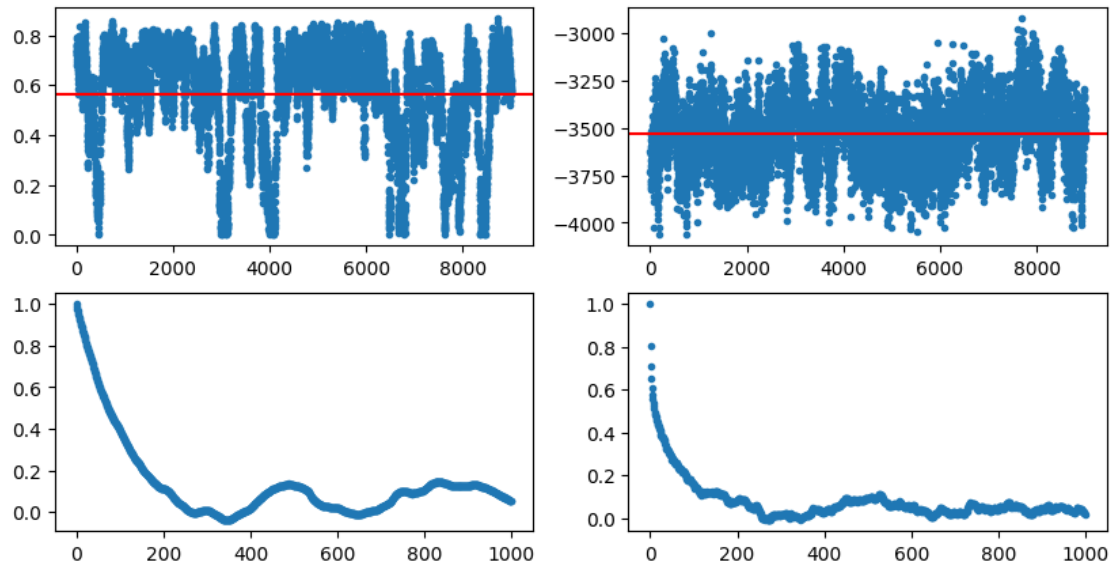
[2]:
```
data = np.loadtxt("../data/analysis1.txt") # with initial phase, 5000 sweep
fig ,ax = plt.subplots(2,2, figsize=(10,5))
spin = data[:,0]
energy = data[:,1]
spin_mean = np.mean(spin[200:])
energy_mean = np.mean(energy[200:])
ax[0,0].plot(spin,'.',label= "Spin")
ax[0,0].axhline(spin_mean,c='r')
ax[0,1].plot(energy,'.',label ="Energie")
ax[0,1].axhline(energy_mean,c='r')
lag = 1000
ax[1,0].plot(autocorr_array(spin,lag),'.')
ax[1,1].plot(autocorr_array(energy,lag),'.')

plt.show()
```



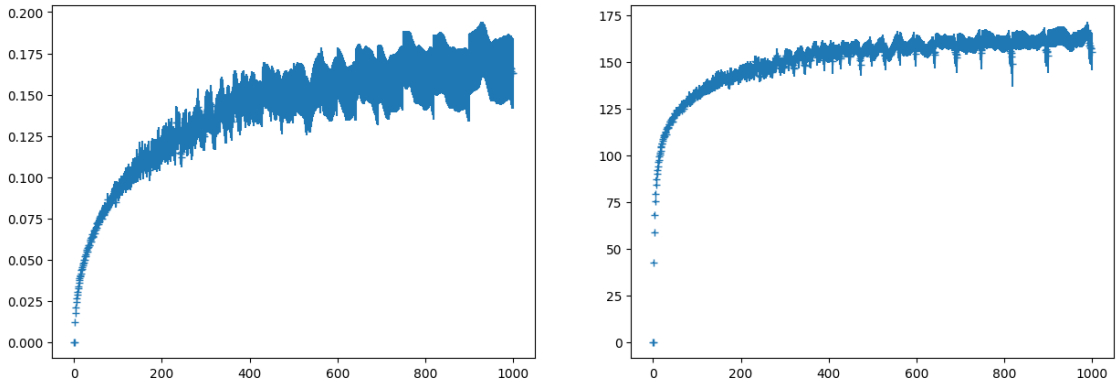
[3]:
```
data = np.loadtxt("../data/analysis2.txt") # with initial phase, 1000 sweep
fig ,ax = plt.subplots(2,2, figsize=(10,5))
spin = data[:,0]
energy = data[:,1]
spin_mean = np.mean(spin[200:])
energy_mean = np.mean(energy[200:])

ax[0,0].plot(spin,'.',label= "Spin")
ax[0,0].axhline(spin_mean,c='r')
ax[0,1].plot(energy,'.',label ="Energie")
ax[0,1].axhline(energy_mean,c='r')
lag = 200
ax[1,0].plot(autocorr_array(spin,lag),'.')
ax[1,1].plot(autocorr_array(energy,lag),'.')

plt.show()
```



[4]:
```
data = np.loadtxt("../data/analysis3.txt") # without initial phase, 10000 sweep␣
  ↪after 1000 sweep ,near Tc
fig ,ax = plt.subplots(2,2, figsize=(10,5))
spin = data[:,0]
energy = data[:,1]
spin_mean = np.mean(spin)
energy_mean = np.mean(energy)

ax[0,0].plot(spin,'.',label= "Spin")
ax[0,0].axhline(spin_mean,c='r')
ax[0,1].plot(energy,'.',label ="Energie")
ax[0,1].axhline(energy_mean,c='r')
lag = 1000
ax[1,0].plot(autocorr_array(spin,lag),'.')
ax[1,1].plot(autocorr_array(energy,lag),'.')

plt.show()
```

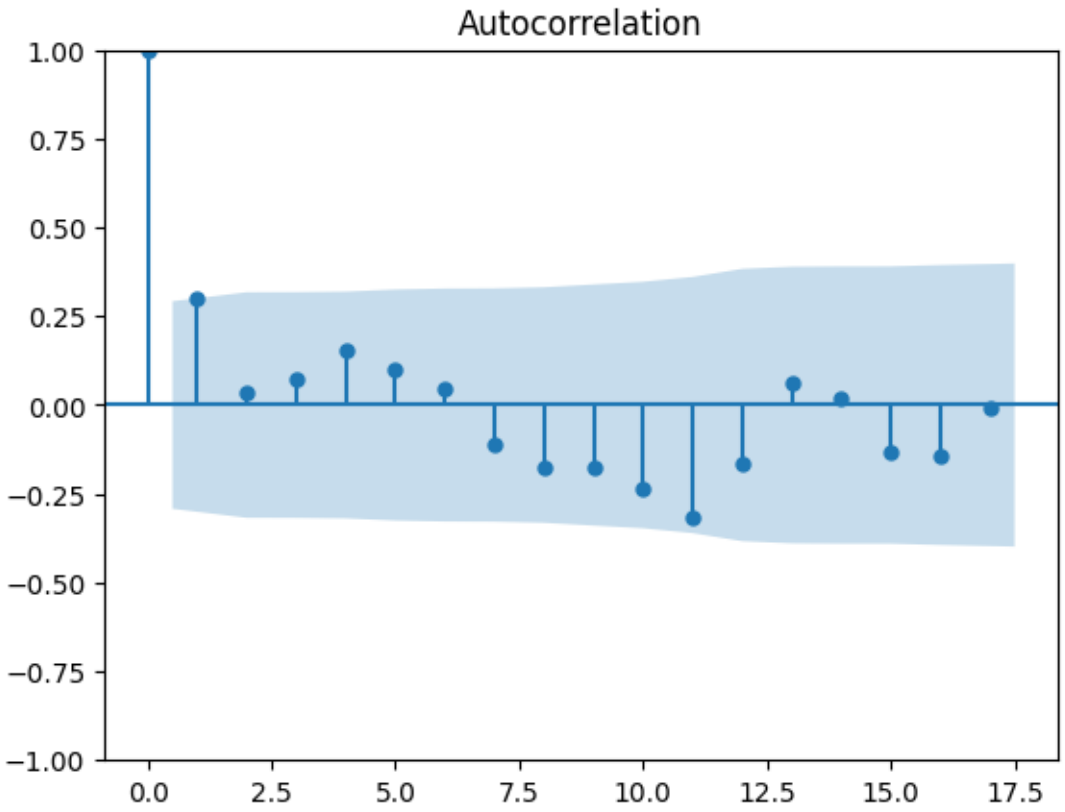


[5]:
```
# Finde Blocklaenge
fig,ax = plt.subplots(1,2,figsize=(15,5))
length = 1000
x_1,y_1,y_err = data_blocking_error(spin,length)
ax[0].errorbar(x_1,y_1,yerr=y_err,fmt= '+',barsabove=True)
x_1,y_1,y_err = data_blocking_error(energy,length)
ax[1].errorbar(x_1,y_1,yerr=y_err,fmt= '+',barsabove=True)
plt.show()
```

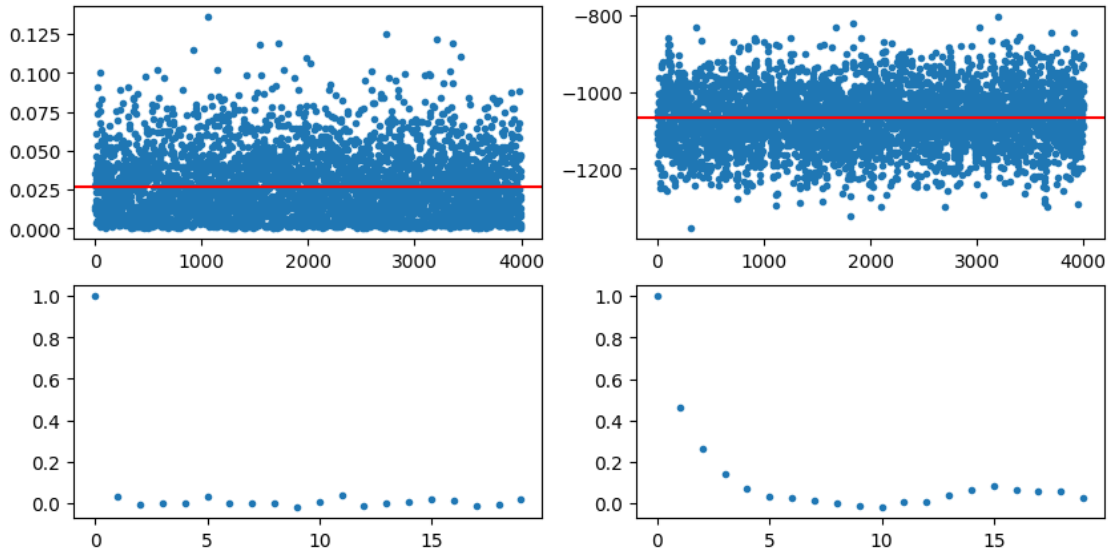


[6]:
```
new_data = data_blocking(spin,200)
plot_acf(new_data)
```

```
plt.show()
```



[9]:
```
data = np.loadtxt("../data/analysis5.txt")
# without initial phase, 5000 sweep after 1000 sweep, heigher temp
fig ,ax = plt.subplots(2,2, figsize=(10,5))
spin = data[:,0]
energy = data[:,1]
spin_mean = np.mean(spin)
energy_mean = np.mean(energy)

ax[0,0].plot(spin,'.',label= "Spin")
ax[0,0].axhline(spin_mean,c='r')
ax[0,1].plot(energy,'.',label ="Energie")
ax[0,1].axhline(energy_mean,c='r')
lag = 20
ax[1,0].plot(autocorr_array(spin,lag),'.')
ax[1,1].plot(autocorr_array(energy,lag),'.')
plt.show()
```



[8]:
```
data = np.loadtxt("../data/analysis6.txt")
x_achse = data[:,0]
spin_mag = data[:,2]
spc_heat = data[:,1]
fig, ax = plt.subplots(1,2,figsize=(15,5))
ax[0].plot(x_achse,spin_mag,'.',label='Spin')
ax[1].plot(x_achse,spc_heat,'.',label='SPC')

ax[0].legend()
ax[1].legend()
plt.show()
```