```python
# Exercise 4.3bc

import numpy as np
import matplotlib.pyplot as plt
import sys

title = '4.3bc'

a = 1.4
b = 0.3

cut_tail = 100

T = 10**4
dt = 5*10**-3
t = np.arange(0,T,dt)

x = np.zeros_like(t,dtype=float)
y = x.copy()

x[0] = (np.random.uniform()-0.5)
y[0] = (np.random.uniform()-0.5)

for t in range(len(x)-1):
    x[t+1] = y[t]+1-a*x[t]**2
    y[t+1] = b*x[t]

x = x[cut_tail:]
y = y[cut_tail:]

q_vals = np.linspace(0,2,3)
epsilon_range = np.linspace(10**-3, 2*10**-2, 10)
fig1, axs = plt.subplots(1,3,figsize=(15,15))
fig2, ax = plt.subplots(figsize=(7,7))
color = ['tab:blue', 'tab:green', 'tab:red']

for q_i in range(len(q_vals)):

    q = q_vals[q_i]
    Iq = np.zeros_like(epsilon_range, dtype=float)

    for epsilon_i in range(len(epsilon_range)):

        epsilon = epsilon_range[epsilon_i]
        N_points = len(x)

        xmax = 1.3
        xmin = -xmax
        ymax = 0.4
        ymin = -ymax

        x_bins = np.linspace(xmin, xmax, int((xmax-xmin)/epsilon))
        y_bins = np.linspace(ymin, ymax, int((ymax-ymin)/epsilon))

        plt.figure()
        histogram = plt.hist2d(x, y, bins=[x_bins, y_bins])
        boxes = histogram[0].copy()
        plt.figure().clear()
        plt.close()
        plt.cla()
        plt.clf()

        for i in range(len(boxes[:,0])):
            for j in range(len(boxes[0,:])):
                if boxes[i,j] != 0:
                    if q == 1:
                        N_k = boxes[i,j]
                        Iq[epsilon_i] += ((N_k/N_points)*np.log(1/(N_k/N_points)))
                    else:
                        N_k = boxes[i,j]
                        Iq[epsilon_i] += (N_k/N_points)**q


    x_axis = np.log(1/epsilon_range)
    axs[q_i].set_xlabel(r'$ln(1/\epsilon)$')
    ax.set_xlabel(r'$ln(1/\epsilon)$')
    ax.set_ylabel(r'$ln(1/\epsilon) D_{q}$')

    if q == 1:
        y_axis = Iq
        axs[q_i].set_ylabel(r'$\sum_{k}^{N_{box}} p_{k} ln(1/p_{k})$')
    else:
        y_axis = np.log(Iq)/(1-q)
        axs[q_i].set_ylabel(r'$ln(\sum_{k}^{N_{box}} p_{k}^{q})/(1-q)$')

    coef = np.polyfit(x_axis, y_axis, 1)
    print(coef)
    Dq = coef[0]

    axs[q_i].plot(x_axis, y_axis, color=color[q_i])
    axs[q_i].set_title('$q={}$, $D_{}={}$'.format(int(q),int(q),np.round(Dq,2)))
    axs[q_i].set_box_aspect(1)
    axs[q_i].set_ylim([5,11])
    axs[q_i].set_xlim([np.log(1/epsilon_range[-1]), np.log(1/epsilon_range[0])])

    legend = '$q={}$, $D_{}'.format(int(q),int(q)) + r'\approx' + '{}$'.format(np.round(Dq,2))
    ax.plot(x_axis, y_axis, 'o-', color=color[q_i], label=legend)
    ax.set_title(title)
    ax.set_box_aspect(1)
    ax.set_ylim([5,11])
    ax.set_xlim([np.log(1/epsilon_range[-1]), np.log(1/epsilon_range[0])])

plt.subplots_adjust(wspace=1)
ax.legend(loc='lower right')
fig1.savefig('Dynamical Systems/DS HW4/4.3/'+title+'_v1.png')
fig2.savefig('Dynamical Systems/DS HW4/4.3/'+title+'_v2.png')
plt.show()
```