Listing 1 ”Nadaraya Watson Estimator of rectangular kernel.”

```
import numpy as np
import scipy.stats as stats
import matplotlib.pyplot as plt

from matplotlib import rc

rc('text', usetex=True)

def Nadaraya_Watson(x, Xi, Yi, h):
    k = []
    nw = 0
    for (x_i, y_i) in zip(Xi, Yi):
        u = ( x_i - x ) / h
        k_u = Rect(u)

        nw += y_i * k_u
        k.append(k_u)

    s = sum(k)

    if s == 0:
        return 0
    else:
        return nw / s

def Rect(u):
    if( abs(u) <= 1 ):
        return 0.5
    else:
        return 0

def f(x):
    return 2 * x * x - x + 3

def AddNoise(x):
    xi = stats.norm.rvs(loc=0, scale = 0.3, size=x.size, random_state=0)
    return x + xi

def Example_1_2(Xi, Yi, f_x, h, ax):
    fn_nw = []
    for _x in Xi:
        fn_nw.append(Nadaraya_Watson(_x, Xi, Yi, h))

    ax.plot(Xi, Yi, 'b+')
    ax.plot(Xi, f_x, 'k--')
    ax.plot(Xi, fn_nw, 'r_')

    ax.set_title('h={0:.2f}'.format(h))
    ax.set_xlabel('x', fontsize=13)
    ax.set_ylabel('y', fontsize=13)

def main():

    sz = 100
    x = np.linspace(start=-1, stop=2.1, num = sz)

    f_x = f(x)
    y = AddNoise(f_x)

    fig, ax = plt.subplots(2, 2, sharex="col", sharey="row")

    Example_1_2(x, y, f_x, 5, ax[0,0])
    Example_1_2(x, y, f_x, 1, ax[0,1])
    Example_1_2(x, y, f_x, 0.1, ax[1,0])
    Example_1_2(x, y, f_x, 0.01, ax[1,1])

    fig.suptitle(r'$Y_i = (2X_i^2 -X_i + 3) + \xi_i , \quad \xi_i \sim N(0, 0.3^2), (i = 1, \ldots, 100)$', fontsize
      =16)

    plt.show()

if __name__ == "__main__":
    main()
```