import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

def kernel(point, xmat, k):

m, n = np.shape(xmat)

weights = np.mat(np.eye((m)))

for j in range(m):

diff = point - X[j]

weights[j, j] = np.exp(diff \* diff.T / (-2.0 \* k \*\* 2))

return weights

def localWeight(point, xmat, ymat, k):

wei = kernel(point, xmat, k)

W = (X.T \* (wei \* X)).I \* (X.T \* (wei \* ymat.T))

return W

def localWeightRegression(xmat, ymat, k):

m, n = np.shape(xmat)

ypred = np.zeros(m)

for i in range(m):

ypred[i] = xmat[i] \* localWeight(xmat[i], xmat, ymat, k)

return ypred

# Load data points

data = pd.read\_csv('10-dataset.csv')

bill = np.array(data.total\_bill)

tip = np.array(data.tip)

# Prepare and add 1 in bill

mbill = np.mat(bill)

mtip = np.mat(tip)

m = np.shape(mbill)[1]

one = np.mat(np.ones(m))

X = np.hstack((one.T, mbill.T))

# Set k here

ypred = localWeightRegression(X, mtip, 0.5)

SortIndex = X[:, 1].argsort(0)

xsort = X[SortIndex][:, 0]

fig = plt.figure()

ax = fig.add\_subplot(1, 1, 1)

ax.scatter(bill, tip, color='green')

ax.plot(xsort[:, 1], ypred[SortIndex], color='red', linewidth=5)

plt.xlabel('Total bill')

plt.ylabel('Tip')

plt.show()

# Working with dataset

Dataset = np.genfromtxt('C:/Users/sahyadri/Downloads/normal\_distribution.csv')

# Mean calculations

mean\_third\_row = np.mean(Dataset[2])

mean\_last\_column = np.mean(Dataset[:, -1])

mean\_first\_3x3 = np.mean(Dataset[0:3, 0:3])

# Median calculations

median\_last\_row = np.median(Dataset[-1])

median\_last\_3\_columns = np.median(Dataset[:, -3])

median\_each\_row = np.median(Dataset, axis=1)

# Variance and standard deviation

variance\_first\_two\_in\_last\_row = np.var(Dataset[-1, :2])

std\_dev\_dataset = np.std(Dataset)

# Indexing and slicing

first\_row = Dataset[0]

last\_element = Dataset[-1, -1]

first\_value\_first\_row = Dataset[0, 0]

second\_last\_row\_reversed = Dataset[-2, ::-1]

# Splitting

ver\_splits = np.vsplit(Dataset, (2,))

# Sorting

sorted\_dataset = np.sort(Dataset, axis=0)

index\_sorted = np.argsort(Dataset[0])

Dataset[0] = Dataset[0][index\_sorted]

# Reshaping

reshaped = np.reshape(Dataset, (1, -1))

reshaped\_alternate = Dataset.reshape(-1, 2)

print("Dataset shape:", Dataset.shape)

print("Subset shape:", ver\_splits[0].shape)