import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import stemgraphic

from scipy.special import comb # For combinatorial calculations (see line 209)

from scipy.stats import hypergeom # For dhyper() equivalent (see line 225)

from scipy.stats import binom # For dbinom() equivalent (see line 264)

from scipy.stats import poisson # For dpois() equivalent (see line 304)

from scipy.stats import nbinom # For dnbinom() equivalent (see line 324)

from scipy.stats import geom # For dgeom() equivalent (see line 338)

from scipy.stats import norm # For pnorm() equivalent (see line 349)

from scipy.stats import probplot # For qqnorm() equivalent (see line 367)

from scipy.stats import lognorm # For plnorm() equivalent (see line 429)

from scipy.stats import expon # For pexp() equivalent (see line 438)

from scipy.stats import gamma # For pgamma() equivalent (see line 453)

from scipy.stats import weibull\_min # For pweibull() equivalent (see line 462)

#Problem 1.A

temp = [127, 125, 131, 124, 129, 121, 142, 151, 160, 125, 124, 123,

120, 119, 128, 133, 137, 124, 142, 123, 121, 136, 140, 137,

125, 124, 128, 129, 130, 122, 118, 131, 125, 133, 141, 125, 140, 131, 129, 126]

temp.sort()

np.median(temp)

#128

#Problem 1.B

#Any amount. It will not change the median

#Problem 1.C

np.mean(temp)

#Mean: 129.975

np.std(temp)

#Std Dev:8.8019

#Problem 1.D

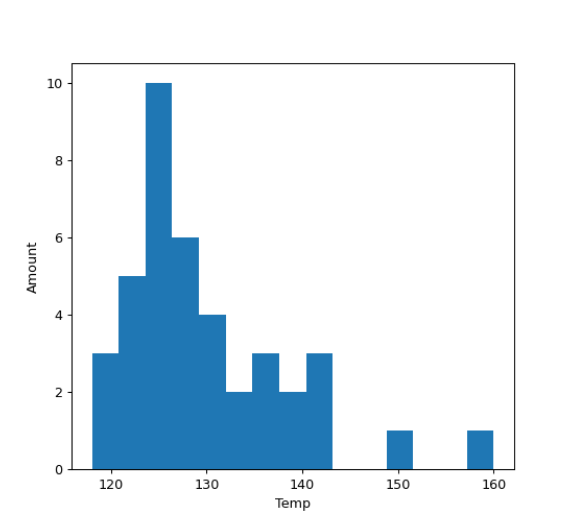
plt.close()

plt.hist(temp, bins=15)

plt.xlabel('Temp')

plt.ylabel('Amount')

plt.show()



#Problem 1.E

plt.close()

stemgraphic.stem\_graphic(temp, scale=10, leaf\_order=True)

plt.show()

