**Chapter 3**

Example 3.4

Code:

import matplotlib.pyplot as plt

# Based on Matplotlib example:

# https://matplotlib.org/3.1.1/gallery/pie\_and\_polar\_charts/pie\_features.html

# Pie chart, where the slices will be ordered and plotted counter-clockwise:

# There are only a few data points so we will enter them manualy. For larger

# data sets, it is better to read data in from a separate file (CSV, etc.)

labels = 'Init. Cutoff', 'Turning', 'Drilling', 'Assembly'

sizes = [80, 170, 73, 37]

fig1, ax1 = plt.subplots()

plt.title

ax1.pie(sizes, labels=labels, autopct='%1.1f%%',

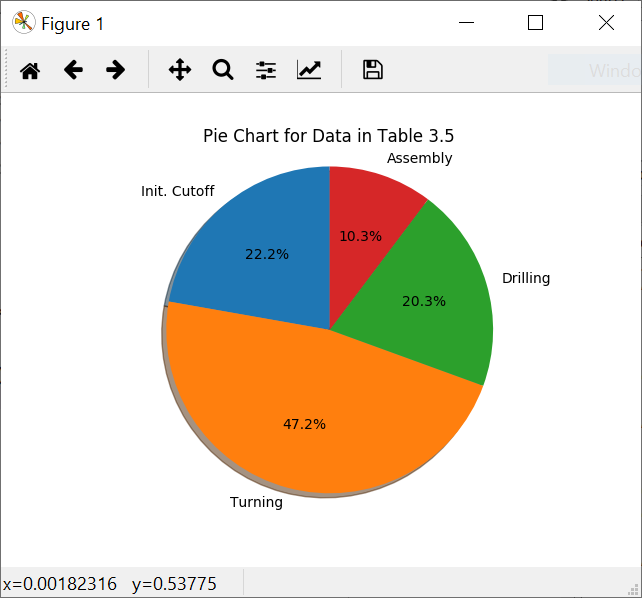
shadow=True, startangle=90)

ax1.axis('equal') # Equal aspect ratio ensures that pie is drawn as a circle.

ax1.set\_title("Pie Chart for Data in Table 3.5")

plt.show()

Output:



Example 3.5

Code:

# Based on Matplotlib example:

# https://matplotlib.org/3.1.1/gallery/lines\_bars\_and\_markers/bar\_stacked.html

# There are only a few data points so we will enter them manualy. For larger

# data sets, it is better to read data in from a separate file (CSV, etc.)

import numpy as np

import matplotlib.pyplot as plt

N = 6

newhires = (128, 245, 130, 154, 152, 165)

ind = np.arange(N) # the x locations for the groups

width = 0.35 # the width of the bars: can also be len(x) sequence

p1 = plt.bar(ind, newhires, width)

plt.ylabel('Hires')

plt.xlabel('Year')

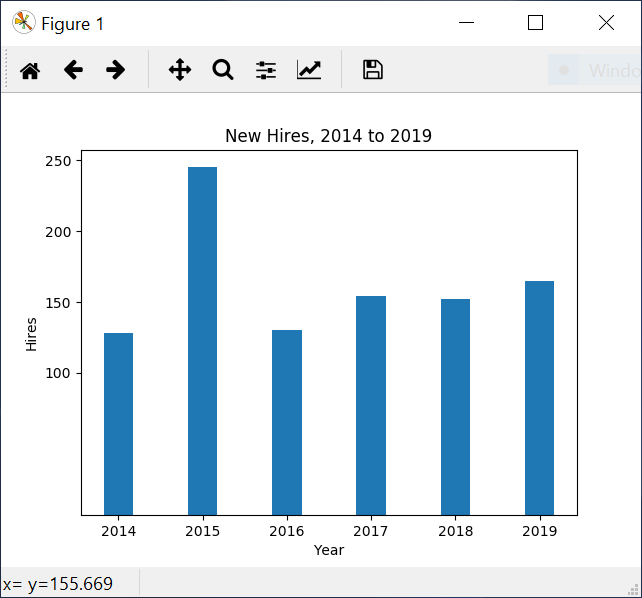
plt.title('New Hires, 2014 to 2019')

plt.xticks(ind, ('2014', '2015', '2016', '2017', '2018', '2019'))

plt.yticks(np.arange(100, 300, 50))

plt.show()

Output:



Example 3.7

Code:

# Based on Matplotlib example:

# https://matplotlib.org/3.1.1/gallery/lines\_bars\_and\_markers/barchart.html

# There are only a few data points so we will enter them manualy. For larger

# data sets, it is better to read data in from a separate file (CSV, etc.)

import matplotlib

import matplotlib.pyplot as plt

import numpy as np

labels = ['A', 'B', 'C', 'D', 'E']

plant\_i = [14, 13, 9, 7, 7]

plant\_ii = [12, 18, 12, 5, 8]

x = np.arange(len(labels)) # the label locations

width = 0.35 # the width of the bars

fig, ax = plt.subplots()

rects1 = ax.bar(x - width/2, plant\_i, width, label='Plant I')

rects2 = ax.bar(x + width/2, plant\_ii, width, label='Plant II')

# Add some text for labels, title and custom x-axis tick labels, etc.

ax.set\_ylabel('Defects')

ax.set\_xlabel('Defect Type')

ax.set\_title('Defect Type by Plant')

ax.set\_xticks(x)

ax.set\_xticklabels(labels)

ax.legend()

def autolabel(rects):

"""Attach a text label above each bar in \*rects\*, displaying its height."""

for rect in rects:

height = rect.get\_height()

ax.annotate('{}'.format(height),

xy=(rect.get\_x() + rect.get\_width() / 2, height),

xytext=(0, 3), # 3 points vertical offset

textcoords="offset points",

ha='center', va='bottom')

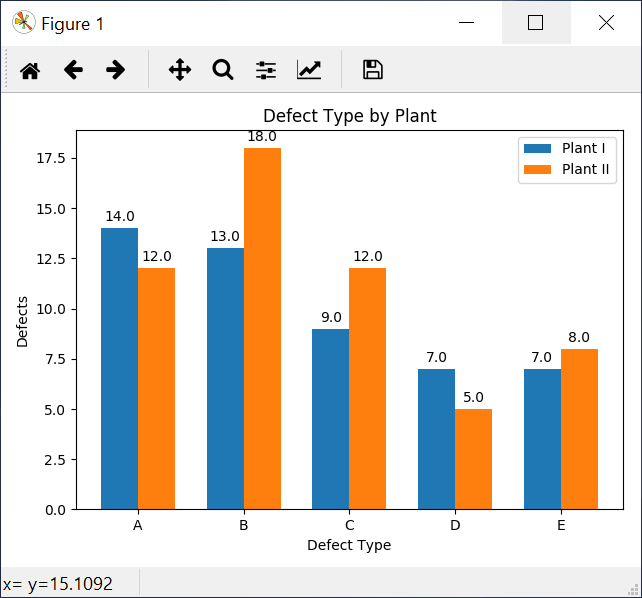
autolabel(rects1)

autolabel(rects2)

fig.tight\_layout()

plt.show()

Output:



Example 3.8 (Histogram)

Code:

# Based on Matplotlib example:

# https://matplotlib.org/3.1.1/gallery/lines\_bars\_and\_markers/barchart.html

# There are only a few data points so we will enter them manualy. For larger

# data sets, it is better to read data in from a separate file (CSV, etc.)

import numpy as np

import matplotlib.mlab as mlab

import matplotlib.pyplot as plt

x=[72, 88, 65, 68, 68, 75, 87, 79, 89, 79,

65, 76, 81, 84, 67, 82, 61, 89, 85, 90,

67, 68, 82, 85, 79, 65, 79, 74, 81, 82]

mu = np.average(x)

sigma = np.std(x)

bins = 6 # Experiment with the number of bins for best visualization

# the histogram of the data

n, bins, patches = plt.hist(x, bins, normed=0, facecolor='green', alpha=0.75)

# Set normed to 1 to normalize to construct a relative frequency histogram.

plt.xlabel('Number of Fuel Pumps')

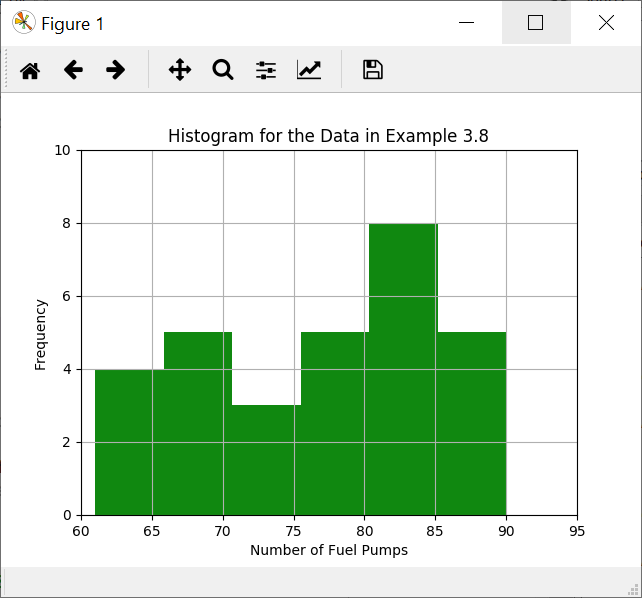
plt.ylabel('Frequency')

plt.title('Histogram for the Data in Example 3.8')

plt.axis([60, 95, 0, 10])

plt.grid(True)

Output:



Example 3.9 – Line graph

Code:

# Line charats are fairly simple, see:

# https://matplotlib.org/3.1.1/tutorials/introductory/pyplot.html

# for some examples.

# There are only a few data points so we will enter them manualy. For larger

# data sets, it is better to read data in from a separate file (CSV, etc.)

import numpy as np

import matplotlib.pyplot as plt

month=np.arange(1, 13, 1) # start is inclusive, end is exclusive (hence 13)

vaccines=[40, 30, 21, 10, 5, 2, 3, 5, 25, 45, 43, 48]

plt.figure(1)

plt.title('line graph for Example 3.9')

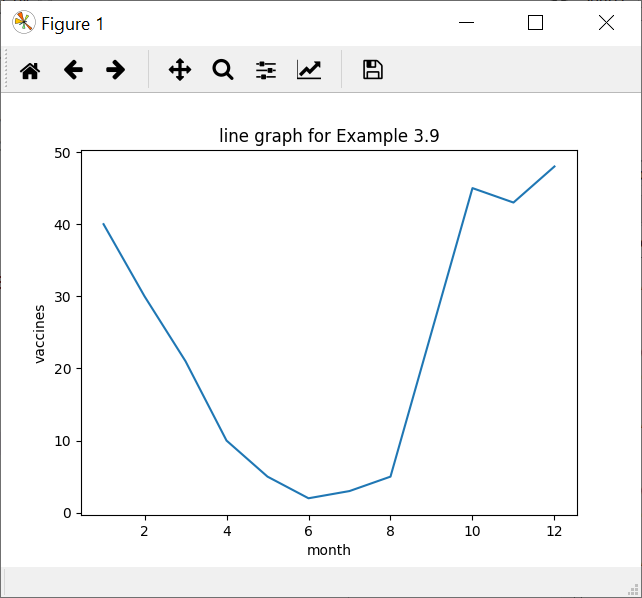
plt.plot(month, vaccines)

plt.xlabel('month')

plt.ylabel('vaccines')

plt.show()

Output:



Example 3.10 – Scatter Plot

Code:

# Scatter plots are fairly simple, see:

# https://matplotlib.org/3.1.1/api/\_as\_gen/matplotlib.pyplot.scatter.html

# for details.

# There are only a few data points so we will enter them manualy. For larger

# data sets, it is better to read data in from a separate file (CSV, etc.)

import matplotlib.pyplot as plt

cholesterol=[195, 180, 220, 160, 200, 220, 200, 183, 139, 155,

153, 164, 171, 143, 159, 167, 162, 165, 178, 145,

245, 198, 156, 175, 171, 167, 142, 187, 158, 142]

systolic=[130, 128, 138, 122, 140, 148, 142, 127, 116, 123,

119, 130, 128, 120, 121, 124, 118, 121, 124, 115,

145, 126, 122, 124, 117, 122, 112, 131, 122, 120]

plt.figure(1)

plt.title('Scatter Plot for Example 3.10')

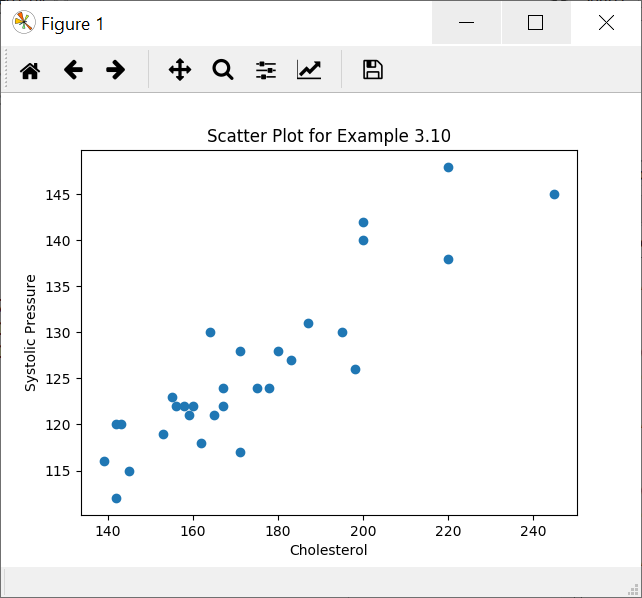
plt.scatter(cholesterol, systolic)

plt.xlabel('Cholesterol')

plt.ylabel('Systolic Pressure')

plt.show()

Output:



Example 3.23 – Statistical Parameters

Code:

# Statistical parameters using NumPy and SciPy's stats module

# There are only a few data points so we will enter them manualy. For larger

# data sets, it is better to read data in from a separate file (CSV, etc.)

import numpy as np

from scipy import stats

salaries=[58, 63, 65, 68, 64, 66, 68, 72, 73, 79, 82, 83, 86, 88, 89]

mean=np.mean(salaries)

median=np.median(salaries)

mode=stats.mode(salaries)[0][0]

variance=np.var(salaries)

stdev=np.std(salaries)

# First quartile (Q1)

q1 = np.percentile(salaries, 25, interpolation = 'midpoint')

# Third quartile (Q3)

q3 = np.percentile(salaries, 75, interpolation = 'midpoint')

# Interquaritle range (IQR)

iqr = q3 - q1

print("Salaries: " + str(salaries))

print("mean: " + str(mean))

print("median: " + str(median))

print("mode: " + str(mode))

print("variance: " + str(variance))

print("Standard Deviation: " + str(stdev))

print("Interquartile Range: " + str(iqr))

Output:

Salaries: [58, 63, 65, 68, 64, 66, 68, 72, 73, 79, 82, 83, 86, 88, 89]

mean: 73.6

median: 72.0

mode: 68

variance: 95.44

Standard Deviation: 9.76933979346

Interquartile Range: 17.0

Example 3.24 – Box Plots

Code:

# Box Plot using Matplotlib. See:

# https://matplotlib.org/3.1.1/api/\_as\_gen/matplotlib.pyplot.boxplot.html

# for details and examples

# There are only a few data points so we will enter them manualy. For larger

# data sets, it is better to read data in from a separate file (CSV, etc.)

import matplotlib.pyplot as plt

data = [85, 80, 88, 95, 115, 110, 105, 104, 89, 97, 96, 140, 75, 79, 99]

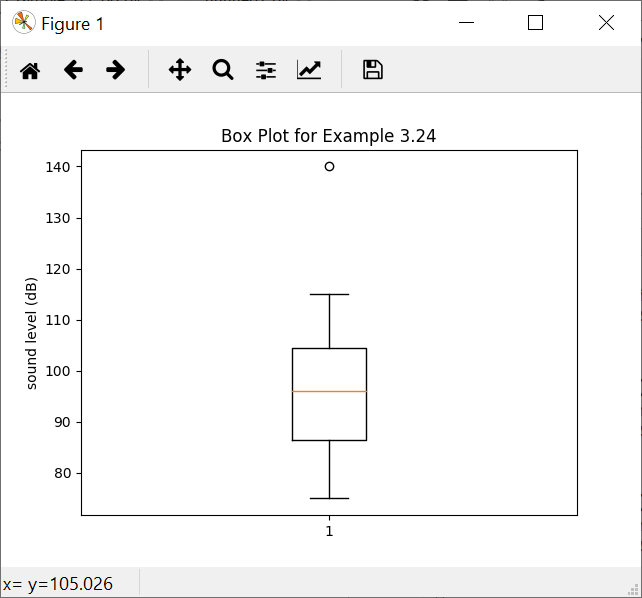
plt.figure()

plt.title('Box Plot for Example 3.24')

plt.ylabel('sound level (dB)')

plt.boxplot(data)

Output:



Example 3.25 – Binomial distribution

Code:

from scipy.stats import binom

success = binom.pmf(k=[0,1,2,3,4,5], n=5, p=0.8)

for x in range(len(success)):

print('X=' + str(x) + ': ' + str(round(success[x], 5)))

Output:

X=0: 0.00032

X=1: 0.0064

X=2: 0.0512

X=3: 0.2048

X=4: 0.4096

X=5: 0.32768

Example 3.27 – Hypergeometric Distribution

Code:

from scipy.stats import hypergeom

# Note that the symbols for M, n, N are not universally standardized.

# scipy's hypergeom uses:

# M for total population

# n for sample size

# N for sub-population of interest

defectives = hypergeom.pmf(k=[0,1,2,3,4,5], M=20, n=10, N=5)

for x in range(len(defectives)):

print('X=' + str(x) + ': ' + str(round(defectives[x], 5)))

Output:

X=0: 0.01625

X=1: 0.13545

X=2: 0.3483

X=3: 0.3483

X=4: 0.13545

X=5: 0.01625

Example 3.31 – Poisson Distribution

Code:

from scipy.stats import poisson

accident\_prob = poisson.pmf(5, 2)

print('Accident Probability:' + str(round(accident\_prob, 5)))

Output:

Accident Probability: 0.03609

Example 3.33 – Normal Distribution

Code:

from scipy.stats import norm

mean = 18

stdev = 1.5

usl = 21

lsl = 15

total = norm.cdf(usl, mean, stdev) - norm.cdf(lsl, mean, stdev)

print(str(round(total \* 100, 4)) + ' Percent') # Convert to percent

Output:

95.45 Percent

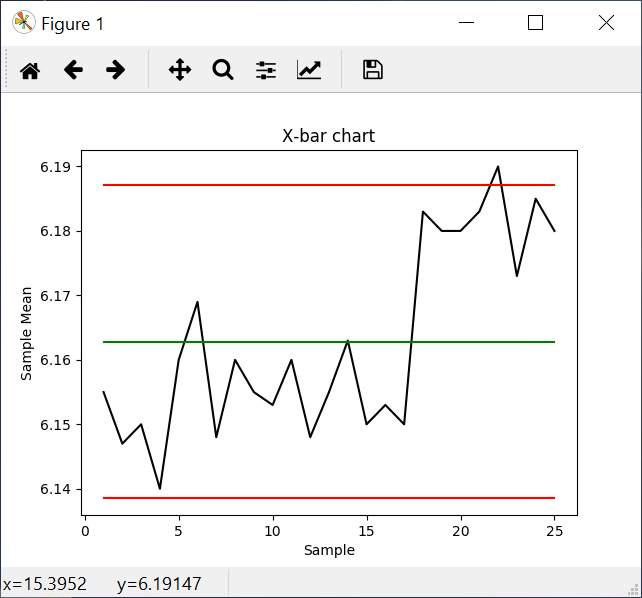
**Chapter 5**

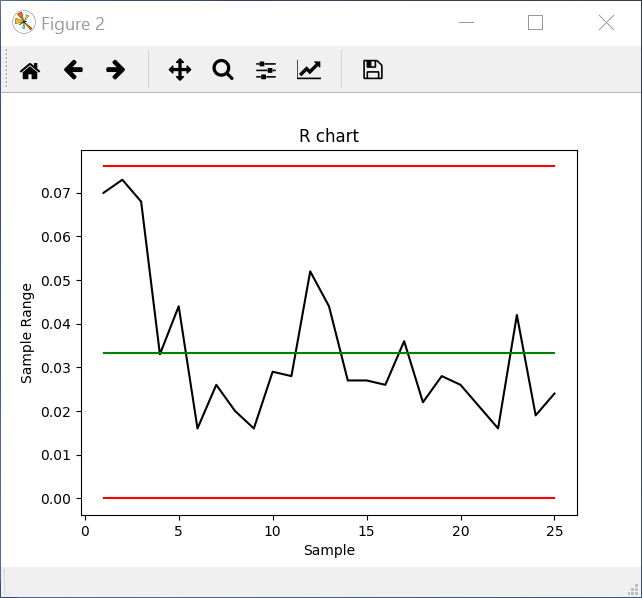
Example 5.4

Code:

See <https://github.com/thorenscientific/py/sqc>

Output:



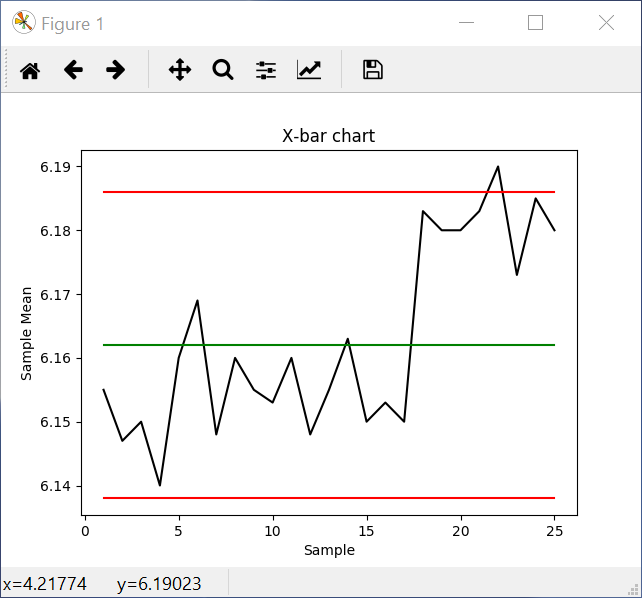


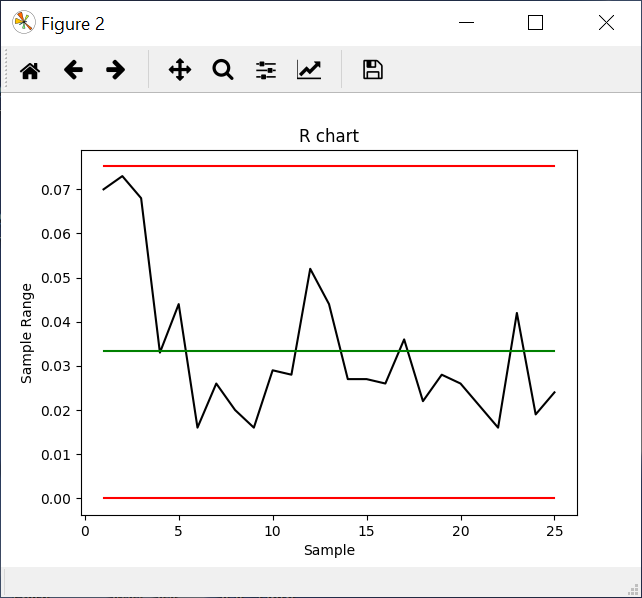
Example 5.5

Code:

See <https://github.com/thorenscientific/py/sqc>

Output:



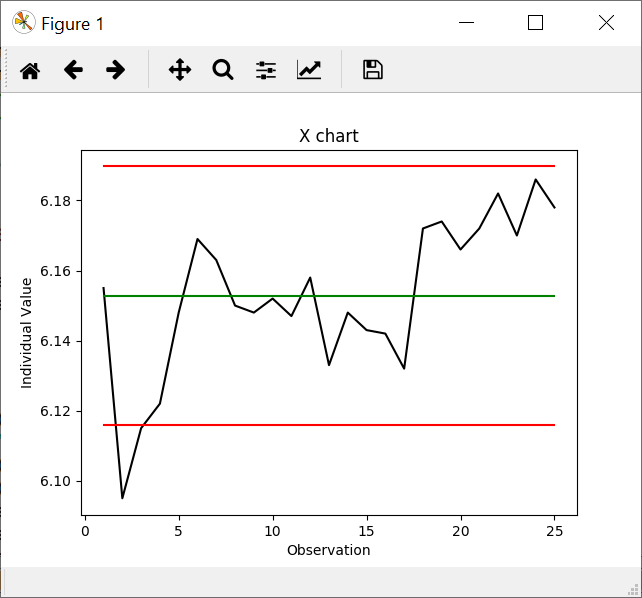


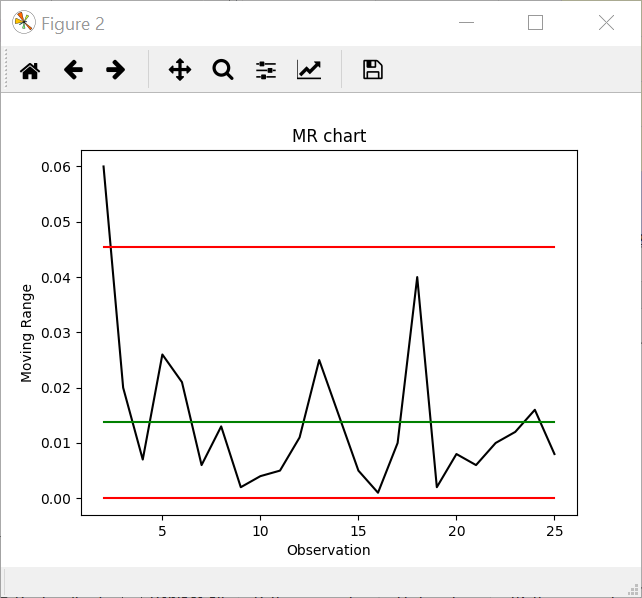
Example 5.6

Code:

See <https://github.com/thorenscientific/py/sqc>

Output:



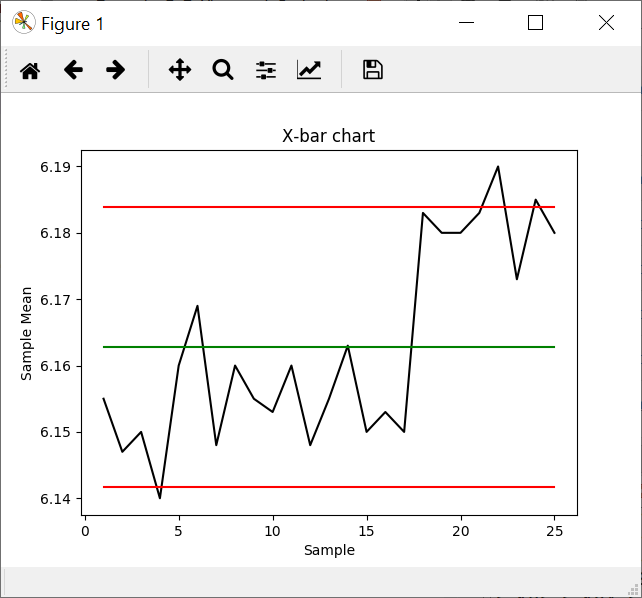


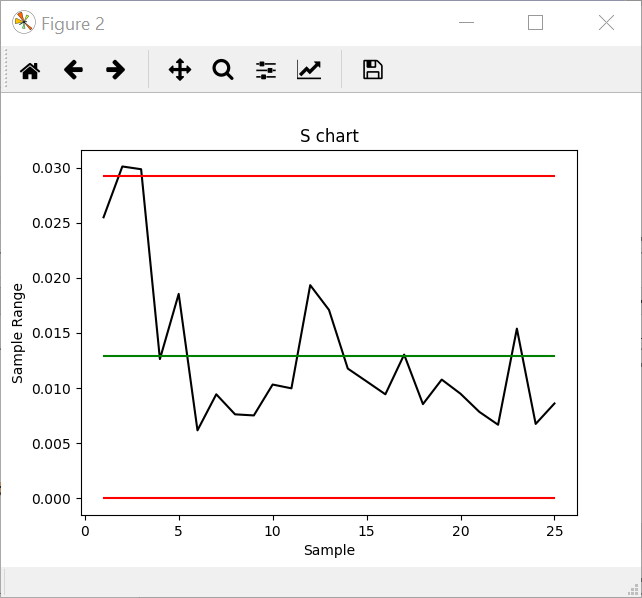
Example 5.7

Code:

See <https://github.com/thorenscientific/py/sqc>

Output:



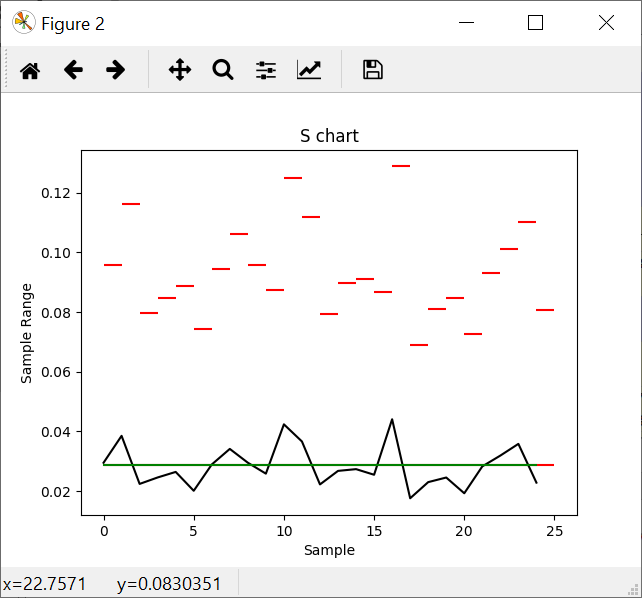


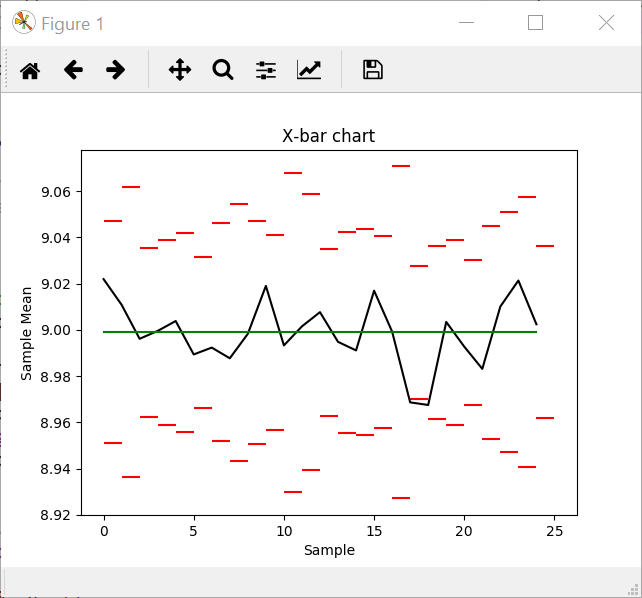
Example 5.8

Code:

See <https://github.com/thorenscientific/py/sqc>

Output:



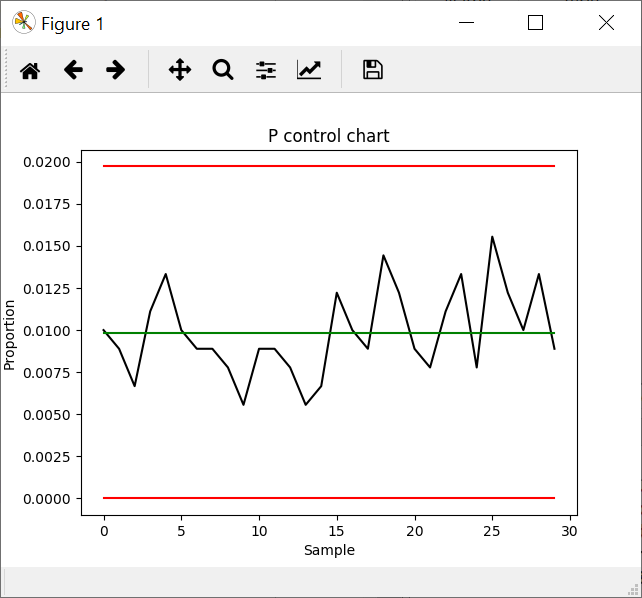


Example 6.1

Code:

See <https://github.com/thorenscientific/py/sqc>

Output:

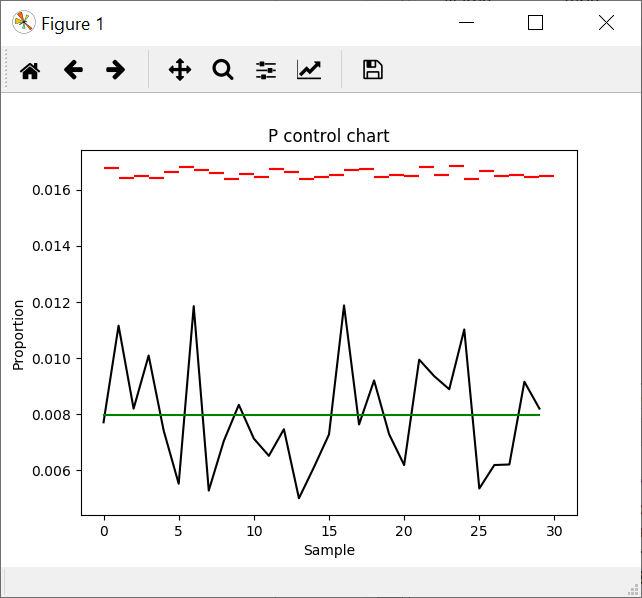


Example 6.2

Code:

See <https://github.com/thorenscientific/py/sqc>

Output:

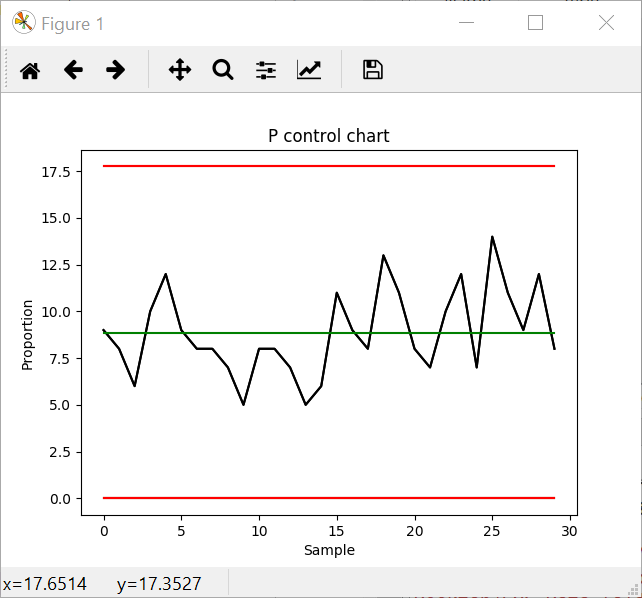


Example 6.3

Code:

See <https://github.com/thorenscientific/py/sqc>

Output:

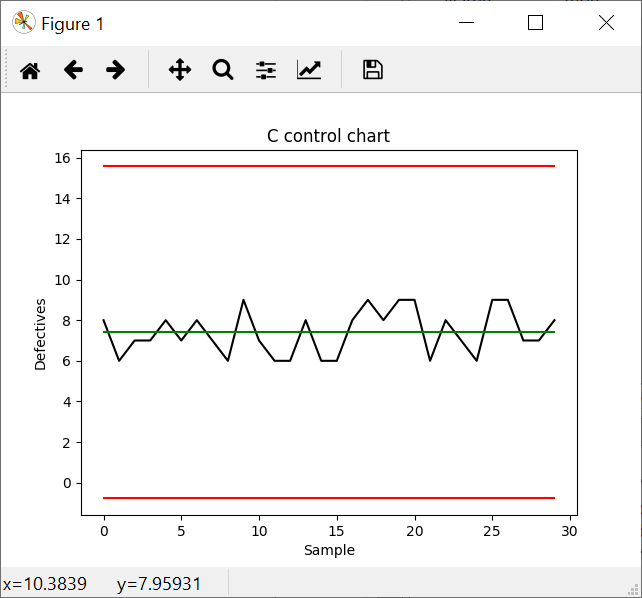


Example 6.4

Code:

See <https://github.com/thorenscientific/py/sqc>

Output:

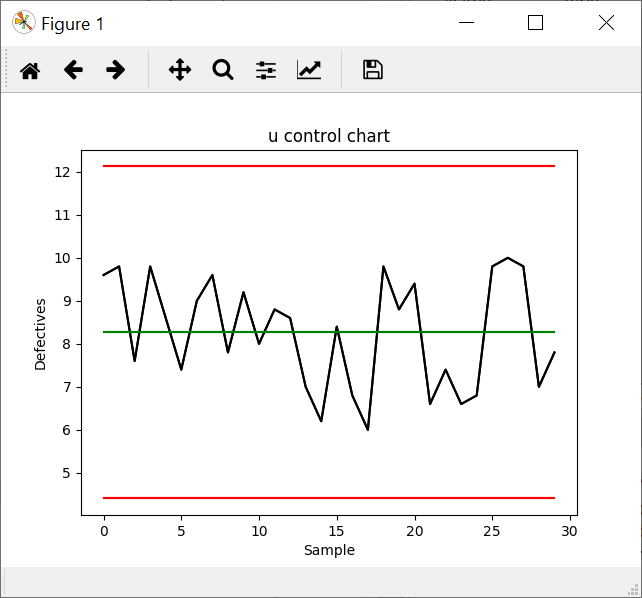


Example 6.5

Code:

See <https://github.com/thorenscientific/py/sqc>

Output:



Example 6.6

Code:

See <https://github.com/thorenscientific/py/sqc>

Output:

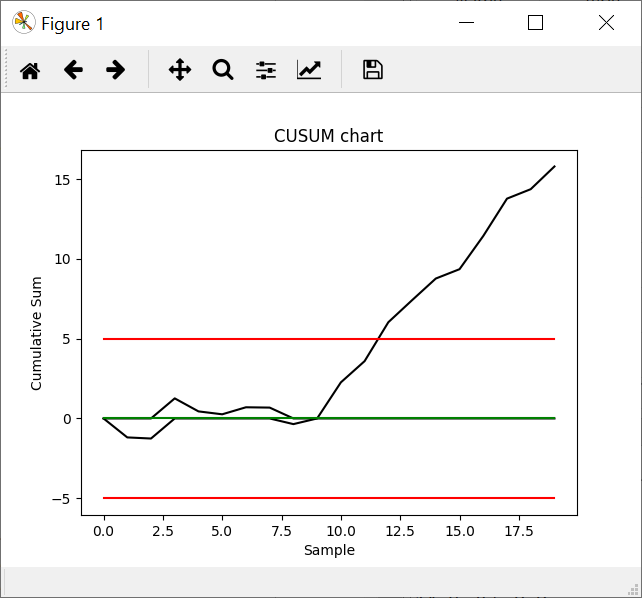
Example 7.1

Code:

See <https://github.com/thorenscientific/py/sqc>

Output:

(Figure)



(console)

Read in 20 samples from data file

CUSUM table

Sample Number, SI+, SI-

0,0,0

1,0.0,-1.1875

2,0.0,-1.2575

3,1.2575,0.0

4,0.4425,0.0

5,0.26,0.0

6,0.7,0.0

7,0.68,0.0

8,0.0,-0.355

9,0.0,0.0

10,2.26,0.0

11,3.59,0.0

12,6.0275,0.0

13,7.4075,0.0

14,8.765,0.0

15,9.35,0.0

16,11.43,0.0

17,13.77,0.0

18,14.3625,0.0

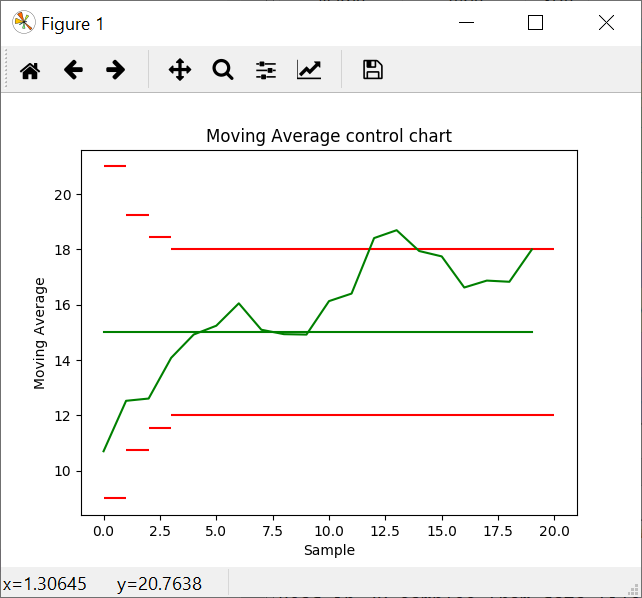
19,15.785,0.0

Example 7.7

Code:

See <https://github.com/thorenscientific/py/sqc>

Output:

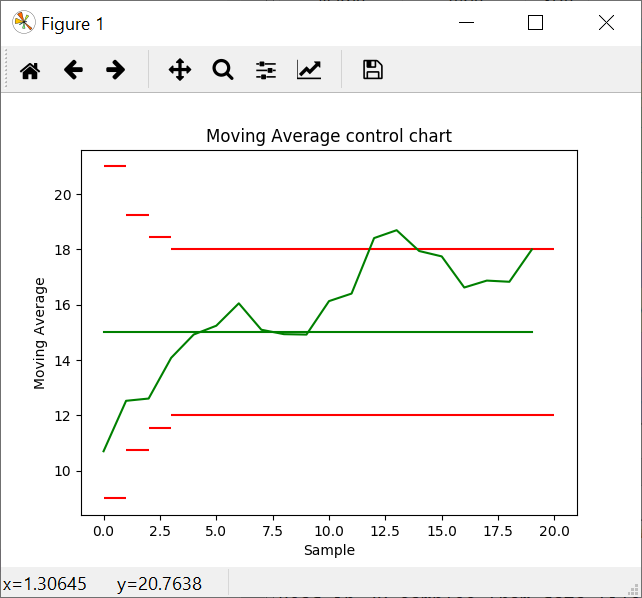


Example 7.8

Code:

See <https://github.com/thorenscientific/py/sqc>

Output:

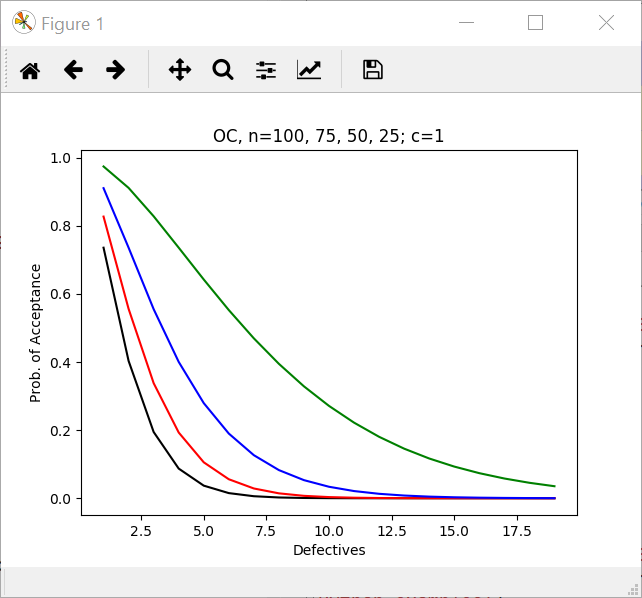


Example 9.1

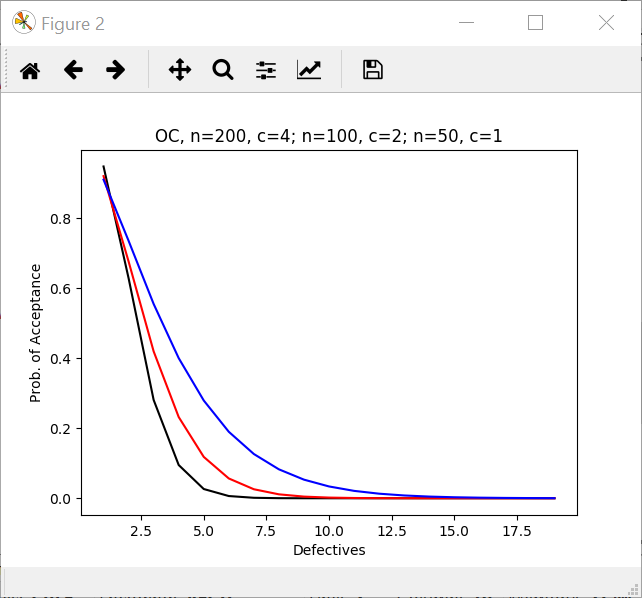
Code:

See <https://github.com/thorenscientific/py/sqc>

Output: (9.1(a))



Output: (9.1(b))



Output: (9.1(b))

