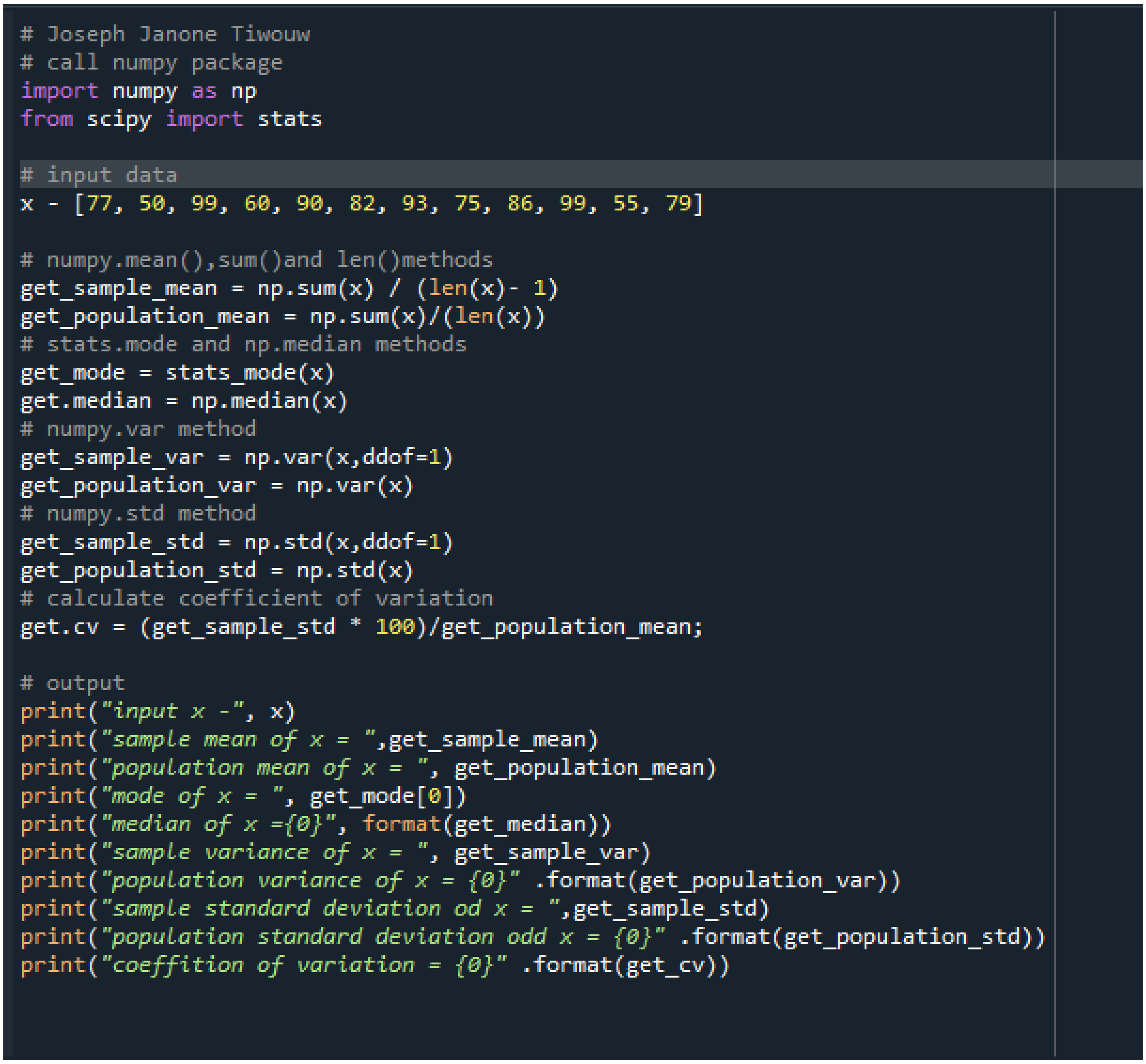
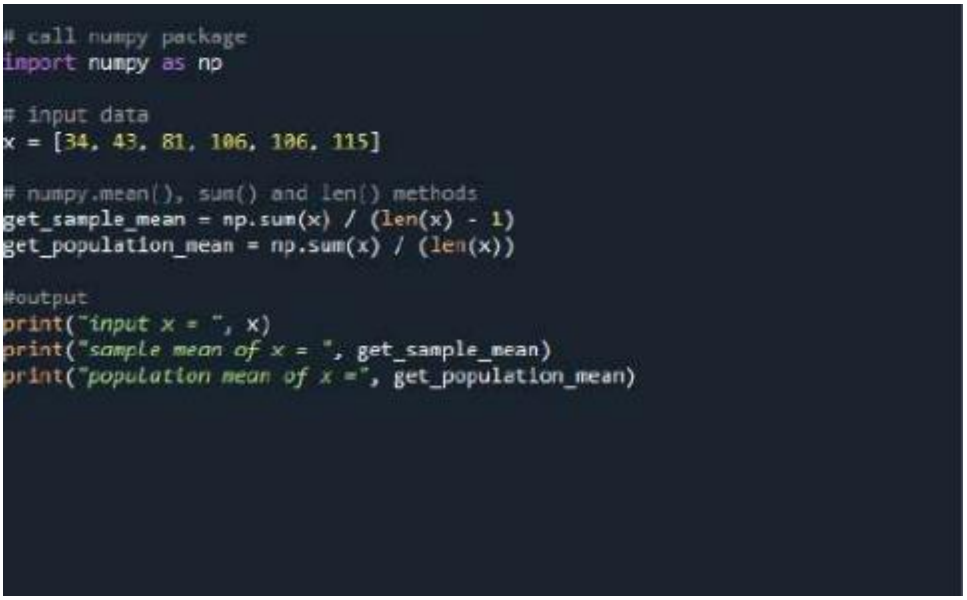
Joseph Janone Tiwouw

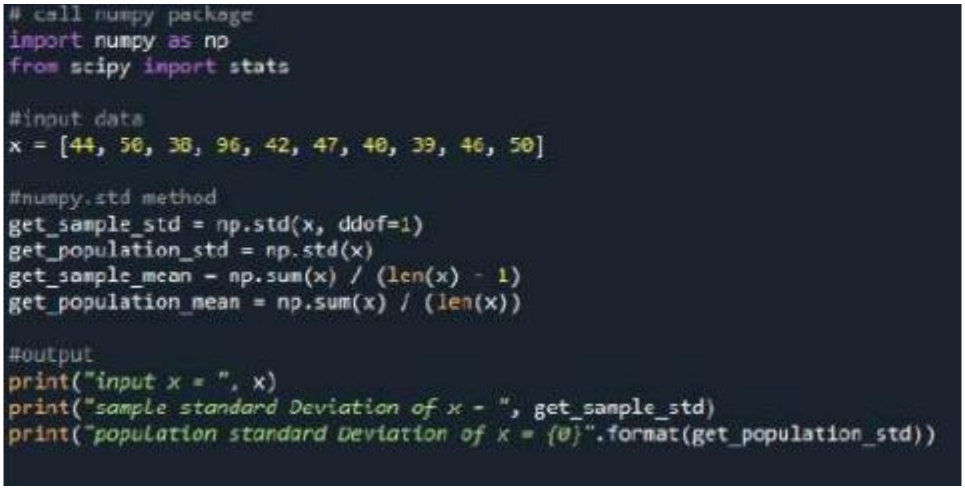
Exercise Lecture 4 & 5



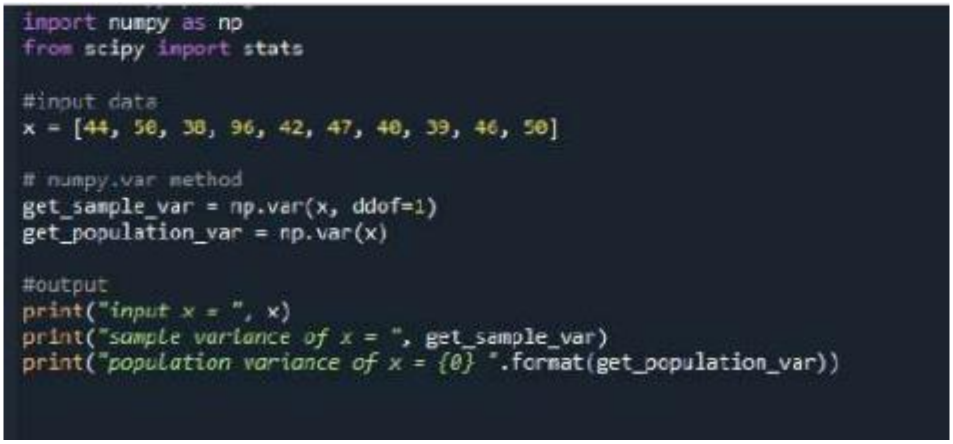
```
# Joseph Janone Tiwouw
# call numpy package
import numpy as np
from scipy import stats

# input data
x - [77, 50, 99, 60, 90, 82, 93, 75, 86, 99, 55, 79]

# numpy.mean(),sum()and len()methods
get_sample_mean = np.sum(x) / (len(x)- 1)
get_population_mean = np.sum(x)/(len(x))
# stats.mode and np.median methods
get_mode = stats_mode(x)
get.median = np.median(x)
# numpy.var method
get_sample_var = np.var(x,ddof=1)
get_population_var = np.var(x)
# numpy.std method
get_sample_std = np.std(x,ddof=1)
get_population_std = np.std(x)
# calculate coefficient of variation
get.cv = (get_sample_std * 100)/get_population_mean;

# output
print("input x -", x)
print("sample mean of x = ",get_sample_mean)
print("population mean of x = ", get_population_mean)
print("mode of x = ", get_mode[0])
print("median of x ={0}", format(get_median))
print("sample variance of x = ", get_sample_var)
print("population variance of x = {0}" .format(get_population_var))
print("sample standard deviation od x = ",get_sample_std)
print("population standard deviation odd x = {0}" .format(get_population_std))
print("coeffition of variation = {0}" .format(get_cv))
```



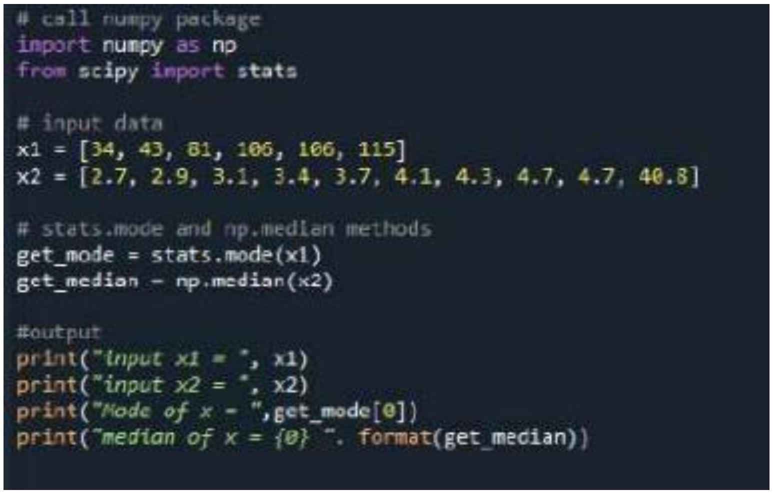
```
# call numpy package
import numpy as np

# input data
x = [34, 43, 81, 106, 106, 115]

# numpy.mean(), sum() and len() methods
get_sample_mean = np.sum(x) / (len(x) - 1)
get_population_mean = np.sum(x) / (len(x))

#output
print("input x = ", x)
print("sample mean of x = ", get_sample_mean)
print("population mean of x =", get_population_mean)
```



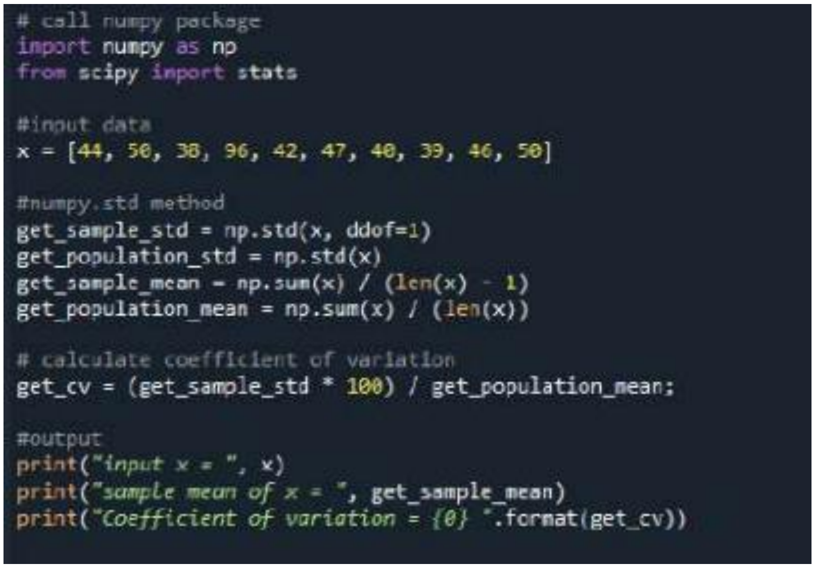
```
# call numpy package
import numpy as np
from scipy import stats

#input data
x = [44, 50, 38, 96, 42, 47, 40, 39, 46, 50]

#numpy.std method
get_sample_std = np.std(x, ddof=1)
get_population_std = np.std(x)
get_sample_mean = np.sum(x) / (len(x) - 1)
get_population_mean = np.sum(x) / (len(x))

#output
print("input x = ", x)
print("sample standard Deviation of x = ", get_sample_std)
print("population standard Deviation of x = {0}".format(get_population_std))
```



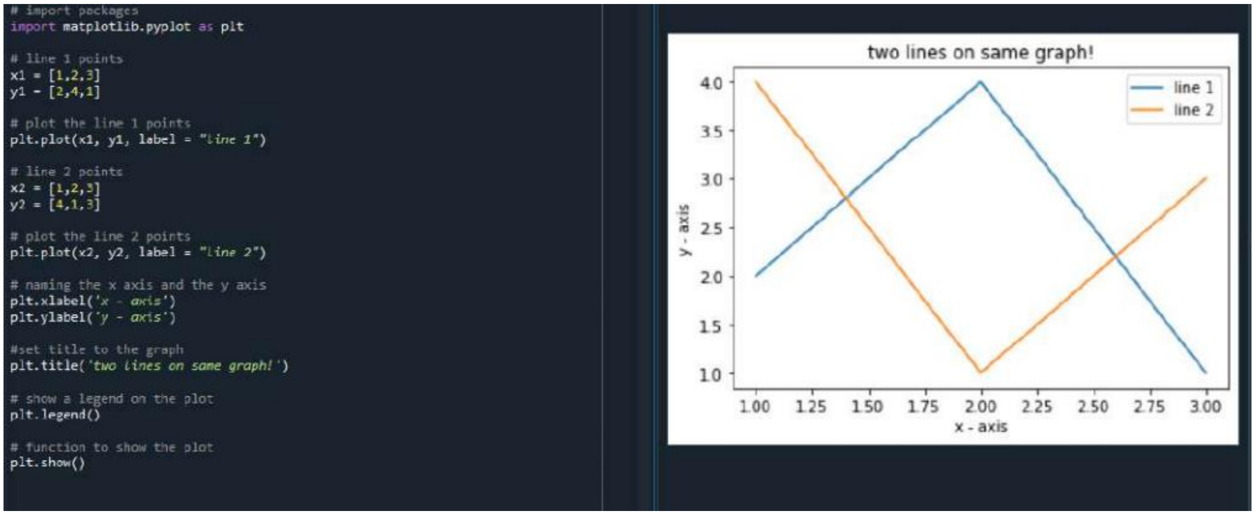
```
import numpy as np
from scipy import stats

#input data
x = [44, 50, 38, 96, 42, 47, 40, 39, 46, 50]

# numpy.var method
get_sample_var = np.var(x, ddof=1)
get_population_var = np.var(x)

#output
print("input x = ", x)
print("sample variance of x = ", get_sample_var)
print("population variance of x = {0} ".format(get_population_var))
```



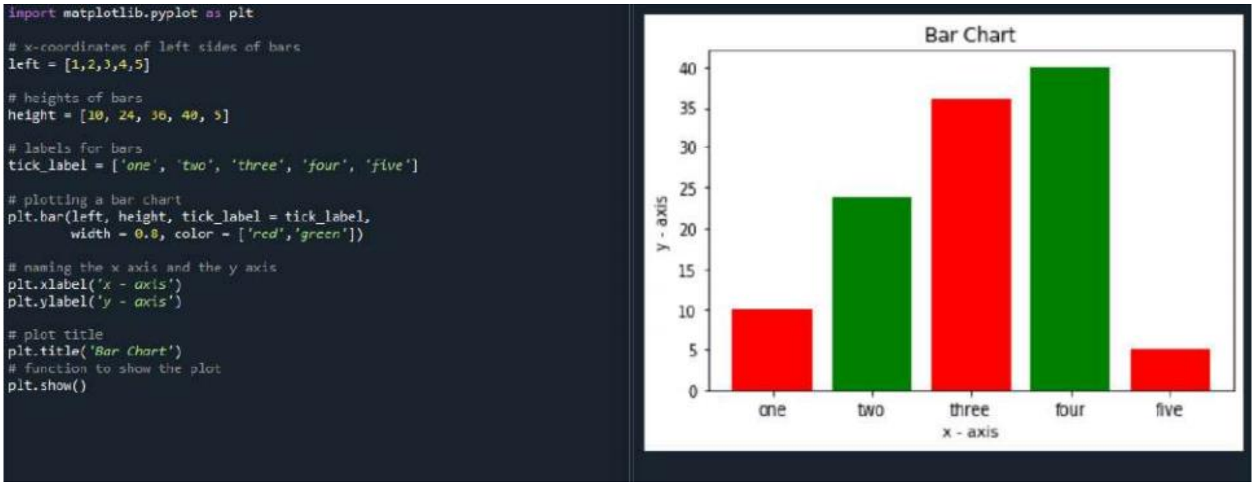
```python
# call numpy package
import numpy as np
from scipy import stats

# input data
x1 = [34, 43, 81, 106, 106, 115]
x2 = [2.7, 2.9, 3.1, 3.4, 3.7, 4.1, 4.3, 4.7, 4.7, 40.8]

# stats.mode and np.median methods
get_mode = stats.mode(x1)
get_median = np.median(x2)

#output
print("input x1 = ", x1)
print("input x2 = ", x2)
print("Mode of x = ",get_mode[0])
print("median of x = {0} ". format(get_median))
```



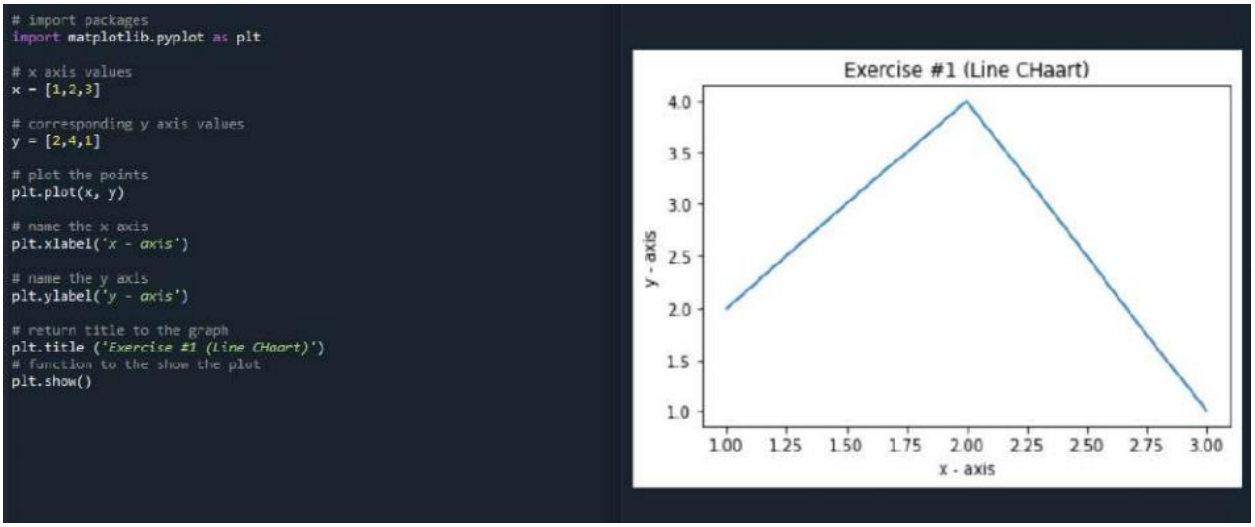
```python
# call numpy package
import numpy as np
from scipy import stats

#input data
x = [44, 50, 38, 96, 42, 47, 40, 39, 46, 50]

#numpy.std method
get_sample_std = np.std(x, ddof=1)
get_population_std = np.std(x)
get_sample_mean = np.sum(x) / (len(x) - 1)
get_population_mean = np.sum(x) / (len(x))

# calculate coefficient of variation
get_cv = (get_sample_std * 100) / get_population_mean;

#output
print("input x = ", x)
print("sample mean of x = ", get_sample_mean)
print("Coefficient of variation = {0} ".format(get_cv))
```



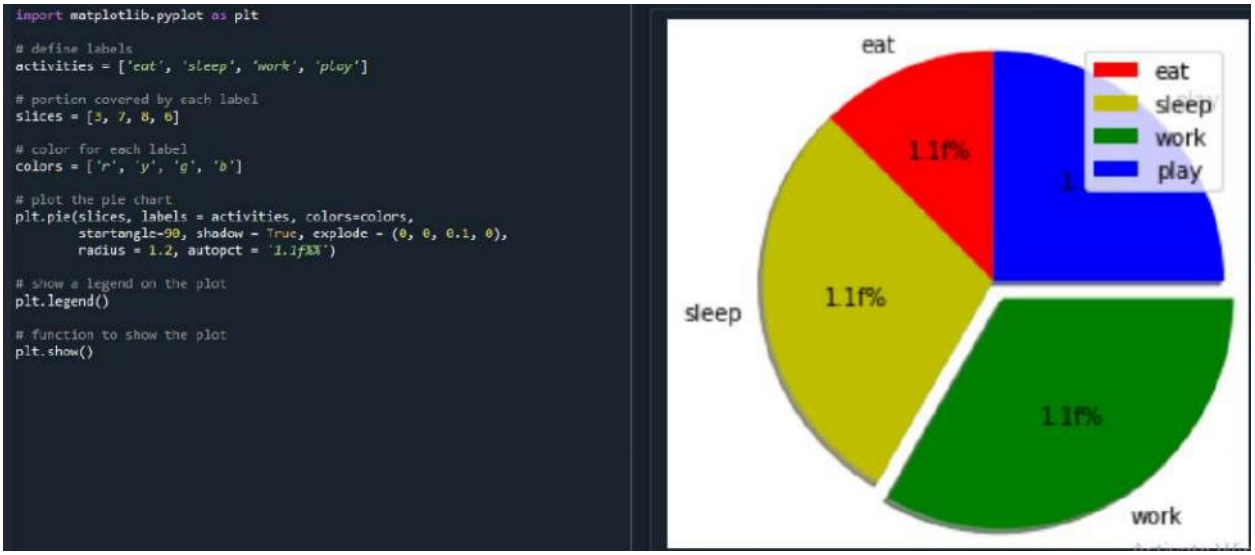
```
# import packages
import matplotlib.pyplot as plt

# line 1 points
x1 = [1,2,3]
y1 = [2,4,1]

# plot the line 1 points
plt.plot(x1, y1, label = "line 1")

# line 2 points
x2 = [1,2,3]
y2 = [4,1,3]

# plot the line 2 points
plt.plot(x2, y2, label = "line 2")

# naming the x axis and the y axis
plt.xlabel('x - axis')
plt.ylabel('y - axis')

#set title to the graph
plt.title('two lines on same graph!')

# show a legend on the plot
plt.legend()

# function to show the plot
plt.show()
```



```
import matplotlib.pyplot as plt

# x-coordinates of left sides of bars
left = [1,2,3,4,5]

# heights of bars
height = [10, 24, 36, 40, 5]

# labels for bars
tick_label = ['one', 'two', 'three', 'four', 'five']

# plotting a bar chart
plt.bar(left, height, tick_label = tick_label,
        width = 0.8, color = ['red','green'])

# naming the x axis and the y axis
plt.xlabel('x - axis')
plt.ylabel('y - axis')

# plot title
plt.title('Bar Chart')
# function to show the plot
plt.show()
```



```
# import packages
import matplotlib.pyplot as plt

# x axis values
x = [1,2,3]

# corresponding y axis values
y = [2,4,1]

# plot the points
plt.plot(x, y)

# name the x axis
plt.xlabel('x - axis')

# name the y axis
plt.ylabel('y - axis')

# return title to the graph
plt.title ('Exercise #1 (Line CHaart)')
# function to the show the plot
plt.show()
```



```
import matplotlib.pyplot as plt

# define labels
activities = ['eat', 'sleep', 'work', 'play']

# portion covered by each label
slices = [3, 7, 8, 6]

# color for each label
colors = ['r', 'y', 'g', 'b']

# plot the pie chart
plt.pie(slices, labels = activities, colors=colors,
        startangle=90, shadow = True, explode = (0, 0, 0.1, 0),
        radius = 1.2, autopct = '1.1f%%')

# show a legend on the plot
plt.legend()

# function to show the plot
plt.show()
```