**Demonstration**

# Number

Year = 2019

#String

School = "PolyU"

#Lists: a list of comma-separated values (items) between square brackets

Week = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']

# Print the variables already defined

print(Year)

print(School)

print(Week)

# Use comma to print multi objectives

print('Data Type of Year: ', type(Year))

print('Data Type of School: ', type(School))

print('Data Type of Week: ', type(Week))

# Use comma to assign multi variables simultaneously

Teacher, Student， Score = 'Linda', 'Leo', 100

print(Teacher, Student, Score)

# create ndarray

import numpy as np

array1 = np.array([(1, 2, 3), (3, 2, 1)])

array2 = np.arange(15)

array3 = array2.reshape(3,5)

array4 = np.zeros((3,4))

print('Type of array1/2/3:', type(array1), type(array2), type(array3), type(array4))

print('array1:\n', array1)

print('array2:\n', array2)

print('array3:\n', array3)

print('array4:\n', array4)

# Create DataFrame

import pandas as pd

import numpy as np

df1 = pd.DataFrame(np.arange(16).reshape((4,4)),index=['a','b','c','d'],columns=['one','two','three','four'])

print('Type of df1: ', type(df1))

print('df1:\n', df1)

# Read data from .csv file

RawData = pd.read\_csv('DailyCoolingLoad.csv')

print('Type of RawData:',type(RawData))

print('Shape of RawData:', RawData.shape)

print('RawData:\n', RawData)

**Case study**

# import modules 导入模块/库

import matplotlib.pyplot as plt

import pandas as pd

import numpy as np

# Data reading 读取数据

Data\_DataFrame = pd.read\_csv('DailyCoolingLoad.csv')

Data\_ndarray = Data\_DataFrame.values

#Visualization 有助于更好地了解并分析数据

colors = ['red', 'green', 'blue']

fig1, ax = plt.subplots()

plt.title('Daily Cooling Load')

plt.xlim((0,144))

plt.ylim((0,25000))

plt.xticks([0,36,72,108,144], ['0','6','12','18','24'])

plt.yticks([0,5000,10000,15000,20000,25000])

for i in range(Data\_ndarray.shape[0]):

ax.plot(Data\_ndarray[i, :], c='black', linewidth=0.2)

ax.set\_xlabel("Time (hr)") #横坐标含义

ax.set\_ylabel('CoolingLoad (kW)') #纵坐标含义

plt.show()

# Hierarchical clusteirng

import scipy.cluster.hierarchy as shc

plt.figure()

plt.title("Hierarchical Clustering Result")

plt.yticks([])

plt.xticks([])

dend = shc.dendrogram(shc.linkage(Data\_ndarray, method='ward'))

from sklearn.cluster import AgglomerativeClustering

cluster = AgglomerativeClustering(n\_clusters=2, affinity='euclidean', linkage='ward')

Results = cluster.fit\_predict(Data\_ndarray)

print('Hierarchical Clustering Results:\n', Results)

# Result visualization

colors = ['red', 'green', 'blue']

fig, ax = plt.subplots()

for i in range(Data\_ndarray.shape[0]):

c=Results[i]

ax.plot(Data\_ndarray[i, :], c=colors[c], linewidth=0.2)

plt.title('Hierarchical Clustering')

plt.xlabel("Time(hr)") #横坐标含义

plt.ylabel('Cooling Load(kW)') #纵坐标含义

plt.xlim((0,144))

plt.ylim((0,25000))

plt.xticks([0,36,72,108,144], ['0','6','12','18','24'])

plt.yticks([0,5000,10000,15000,20000,25000])