UNIT-4 (Scientific Computing with Python: Scipy)

**MCA - III Semester**

**Subject: Data Science & Analytics using Python**

**Unit-IV Assignment**

**Date of Release: 30/11/2022**

**Last Date of Submission: 07/12/2022**

**Part-A 10X1=10**

Q.1 (CO4) SciPy stands for?

A. science Python

B. source Python

C. significant Python

D. scientific Python

Q.2 (CO4) SciPy Original author is?

A. Guido van Rossum

B. Travis Oliphant

C. Wes McKinney

D. Jim Hugunin

Q.3 (CO4) Which of the following is not correct sub-packages of SciPy?

A. scipy.cluster

B. scipy.source

C. scipy.interpolate

D. scipy.signal

Q.4 (CO4) The number of axes is called as \_\_\_\_\_.

A. object

B. Vectors

C. rank

D. matrices

Q.5 (CO4) Which of the following is true?

A. By default, all the NumPy functions have been available through the SciPy namespace

B. There is no need to import the NumPy functions explicitly, when SciPy is imported.

C. SciPy is built on top of NumPy arrays

D. All of the above

Q.6 (CO4) What will be output for the following code?

import numpy as np

print np.arange(7)

A. array([0, 1, 2, 3, 4, 5, 6])

B. array(0, 1, 2, 3, 4, 5, 6)

C. [0, 1, 2, 3, 4, 5, 6]

D. [[0, 1, 2, 3, 4, 5, 6]]

Q.7 (CO4) What will be output for the following code?

import numpy as np

print np.linspace(1., 4., 6)

A. array([ 1. , 2.2, 2.8, 3.4, 4. ])

B. array([ 1. , 1.6, 2.8, 3.4, 4. ])

C. array([ 1. , 1.6, 2.2, 2.8, 3.4, 4. ])

D. array([ 1. , 1.6, 2.2, 2.8, 4. ])

Q.8 (CO4) Which of the following code is used to whiten the data?

A. data = numpy.whiten(data)

B. data = whiten(data)

C. data =SciPy.whiten(data)

D. data = data.whiten()

Q.9 (CO4) How to import Constants Package in SciPy?

A. import scipy.constants

B. from scipy.constants

C. import scipy.constants.package

D. from scipy.constants.package

Q.10 (CO4) What is "h" stand for Constant?

A. Newton's gravitational constant

B. Elementary charge

C. Planck constant

D. Molar gas constant

Q.11 (CO4) what is constant defined for Boltzmann constant in SciPy?

A. G

B. e

C. R

D. k

Q.12 (CO4) What is the value of unit milli in SciPy?

A. 0.01

B. 0.1

C. 0.0001

D. 0.001

Q.13 (CO4) What will be output for the following code?

from scipy import linalg

import numpy as np

a = np.array([[3, 2, 0], [1, -1, 0], [0, 5, 1]])

b = np.array([2, 4, -1])

x = linalg.solve(a, b)

print x

A. array([ 2., -2., 9., 6.])

B. array([ 2., -2., 9.])

C. array([ 2., -2.])

D. array([ 2., -2., 9., -9.])

Q.14 (CO4) What will be output for the following code?

from scipy import linalg

import numpy as np

A = np.array([[1,2],[3,4]])

x = linalg.det(A)

print x

A. 2

B. 1

C. -2

D. -1

Q.15 (CO4). In SciPy, determinant is computed using?

A. determinant()

B. SciPy.determinant()

C. det()

D. SciPy.det()

Q.16 (CO4). scipy.linalg always compiled with?

A. BLAS/LAPACK support

B. BLAS/Linalg support

C. Linalg/LAPACK support

D. None of the above

Q.17 (CO4) Which of the following is false?

A. scipy.linalg also has some other advanced functions that are not in numpy.linalg

B. SciPy version might be faster depending on how NumPy was installed.

C. Both A and B

D. None of the above

Q.18 (CO4) The scipy.linalg.solve feature solves the \_\_\_\_\_\_\_.

A. integration problem

B. differentiation problem

C. linear equation

D. All of the above

Q.19 (CO4) What relation is consider between Eigen value (lambda), square matrix (A) and Eign vector(v)?

A. Av = lambda\*v

B. Av =Constant \* lambda\*v

C. Av =10 \* lambda\*v

D. Av != lambda\*v

Q.20 (CO4) What will be output for the following code?

from scipy.special import logsumexp

import numpy as np

a = np.arange(10)

res = logsumexp(a)

print res

A. 10

B. 9.45862974443

C. 9

D. 9.46

**Part-B 5X2=10**

Q.1(CO4) What is the concept of Packages in Scipy in Python?

[SciPy](https://www.mygreatlearning.com/scipy/free-courses?gl_blog_id=22293) is a free and open-source [Python](https://www.mygreatlearning.com/blog/python-tutorial-for-beginners-a-complete-guide/) library used for scientific computing and technical computing. It is a collection of mathematical algorithms and convenience functions built on the NumPy extension of Python. It adds significant power to the interactive Python session by providing the user with high-level commands and classes for manipulating and visualizing data.

Q.2 (CO4) What do you understand by optimization in reference of scipy library in Python?

SciPy optimize provides functions for minimizing (or maximizing) objective functions, possibly subject to constraints. It includes solvers for nonlinear problems (with support for both local and global optimization algorithms), linear programing, constrained and nonlinear least-squares, root finding, and curve fitting.

Q.3 (CO4) Write the name of different types of sub packages in scipy library of python?

scipy.cluster,scipy.constants,,,scipy.fftpack,scipy.integrate,scipy.interpolation,scipy.linalg,scipy.io,scipy.ndimage,scipy.odr,scipy.optimize,scipy.signal,scipy.sparse,scipy.spatial,scipy.special,scipy.stats,scipy.weves.

Q.4(CO4) Write the statement for importing the scipy library in python?

**import scipy as sp**

Q5 (CO4) What do you understand by Scipy library of python?

SciPy is **a scientific computation library that uses NumPy underneath**. SciPy stands for Scientific Python. It provides more utility functions for optimization, stats and signal processing. Like NumPy, SciPy is open source so we can use it freely.

**Part-C 5X6=30**

Q.1 (CO4) Briefly discuss basis of statistics sub package of scipy in python with example.

This sub-module of SciPy is having a large number of probability distributions and a growing library of statistical functions.

1. **import** scipy as sp
2. **import** numpy as np
3. from scipy.stats **import** norm
4. number\_of\_data = 100
5. random\_data\_set = sp.randn(number\_of\_data)
6. print(random\_data\_set.mean())
7. print(sp.median(random\_data\_set))
8. min\_max = np.array([random\_data\_set.min(),random\_data\_set.max()])
9. print(min\_max)
10. sp.stats.describe(random\_data\_set)

**Output:**

0.006283818005153084

-0.03008382588766136

[-2.1865825 2.47537921]

DescribeResult(nobs=100, minmax=(-2.1865824992721987, 2.475379209985273), mean=0.006283818005153084, variance=1.0933102537156147, skewness=0.027561719919920322, kurtosis=-0.6958272633471831)

Q.2 (CO4) Describe concept of Integration sub package and its types in scipy with example.

The **[scipy.integrate](https://docs.scipy.org/doc/scipy/reference/integrate.html" \l "module-scipy.integrate" \o "scipy.integrate)** sub-package provides several integration techniques including an ordinary differential equation integrator.

**import** scipy.integrate **as** integrate

>>> **import** scipy.special **as** special

>>> result **=** integrate**.**quad**(lambda** x**:** special**.**jv**(2.5,**x**),** **0,** **4.5)**

>>> result

*(1.1178179380783249, 7.8663172481899801e-09)*

>>> **from** numpy **import** sqrt**,** sin**,** cos**,** pi

>>> I **=** sqrt**(2/**pi**)\*(18.0/27\***sqrt**(2)\***cos**(4.5)** **-** **4.0/27\***sqrt**(2)\***sin**(4.5)+**sqrt**(2\***pi**)\*** special**.**fresnel**(3/**sqrt**(**pi**))[0])**

>>> I

*1.117817938088701*

>>> print**(**abs**(**result**[0]-**I**))**

*1.03761443881e-11*

Q.3 (CO4) What do you understand by need and use of weave sub package of scipy in python with example

The [scipy.weave](https://docs.scipy.org/doc/scipy-0.16.1/reference/weave.html" \l "module-scipy.weave" \o "scipy.weave) package provides tools for including C/C++ code within in Python code.

**>>> import** **weave**

**>>>** a = 1

**>>>** weave.inline('printf("*%d***\\**n",a);',['a'])

1

Q.4.(CO4) Describe the concept of Optimization sub package with example used in python.

This SciPy sub module provides us the functions for minimizing or maximizing objective functions. It also includes solvers for nonlinear problems, linear programming, root findings, and curve fitting.

**import** numpy **as** np

>>> **from** scipy.optimize **import** minimize

>>> **def** rosen**(**x**):**

... *"""The Rosenbrock function"""*

... **return** sum**(100.0\*(**x**[1:]-**x**[:-1]\*\*2.0)\*\*2.0** **+** **(1-**x**[:-1])\*\*2.0)**

>>> x0 **=** np**.**array**([1.3,** **0.7,** **0.8,** **1.9,** **1.2])**

>>> res **=** minimize**(**rosen**,** x0**,** method**=**'nelder-mead'**,**

... options**={**'xatol'**:** **1e-8,** 'disp'**:** **True})**

*Optimization terminated successfully.*

*Current function value: 0.000000*

*Iterations: 339*

*Function evaluations: 571*

>>> print**(**res**.**x**)**

*[1. 1. 1. 1. 1.]*

Q.5.(CO4) Describe the concept of IO sub package with example used in python.

This SciPy sub-package contains modules, classes and functions to read data from and write data to various file formats such as MATLAB files, IDL files, Matrix Market files, Unformatted Fortran files, Wav sound files, Arff files.

**import** scipy.io as sio

**import** numpy as np

#Save a mat file

vect = np.arange(10)

sio.savemat('array.mat', {'vect':vect})

#Now Load the File

mat\_file\_content = sio.loadmat('array.mat')

print(mat\_file\_content)

**Output:**

{'\_\_header\_\_': b'MATLAB 5.0 MAT-file Platform: nt, Created on: Wed Nov 13 14:38:57 2019', '\_\_version\_\_': '1.0', '\_\_globals\_\_': [], 'vect': array([[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]])}

**Part-D 2X10=20**

Q.1 (CO4) Create a code in python using scipy library to optimize a Minimize equation as given below

Objective function: x1\*1+x1\*x2

Constraints: x1\*x1\*x1+x1\*x2=100

X1\*x1\*x1+x1\*x2>=50

-100<=x1\*x2<=100

Q.2 (CO4) Solve the given Linear equation using scipy of python

X+2y-3z=-3

2x-5y+4zz=13

5x+4y-z=5

import numpy as np

from scipy import linalg

# Creating the input array

a = np.array([[1, 2, -3], [2, -5, 4], [5, 4, -1]])

# Providing the solution Array

b = np.array([[-3], [13], [5]])

# Solve the linear algebra

x = linalg.solve(a, b)

# Printing the result

print(x)

# Checking the result

np.dot(a, x) == b

## Output

[[ 2.]

[-1.]

[ 1.]]