# LAB CYCLE 1

**20MCA241: DATA SCIENCE LAB**

## EXERCISE 1: Introduction to Numpy CO1

1. Write a NumPy program to create an element-wise comparison (greater, greater\_equal, less and less\_equal) of two given arrays.
2. Write a NumPy program to create an array of all the even integers from 30 to 70.
3. Write a NumPy program to create a 3x3 identity matrix.
4. Write a NumPy program to create a vector with values from 0 to 20 and change the sign of the numbers in the range from 9 to 15.
5. Write a NumPy program to create a 5x5 zero matrix with elements on the main diagonal equal to 1, 2, 3, 4, 5.
6. Write a NumPy program to compute sum of all elements, sum of each column and sum of each row of a given array.
7. Write a NumPy program to save a given array to a text file and load it.
8. Write a NumPy program to check whether two arrays are equal (element wise) or not.
9. Write a NumPy program to create a 4x4 array with random values, now create a new array from the said array swapping first and last rows.
10. Write a NumPy program to multiply two given arrays of same size element-by-element.

## EXERCISE 2: Matrix operations (using vectorization) and transformations CO1

Write a Python program to create two matrices (read values from user) and find the following

* 1. Dot Product
  2. Transpose
  3. Trace
  4. Rank
  5. Determinant
  6. Inverse
  7. Eigenvalues and eigenvectors

## EXERCISE 3: Programs using Matplotlib CO1

1. Draw a line in a diagram from position (1, 3) to (2, 10) then to (6, 12) and finally to position (18, 20). (Mark each point with a beautiful green colour and set line colour to red and line style dotted)
2. Draw a plot for the following data:

|  |  |
| --- | --- |
| Temperature in degree Celsius | Sales |
| 12 | 100 |
| 14 | 200 |
| 16 | 250 |
| 18 | 400 |
| 20 | 300 |
| 22 | 450 |
| 24 | 500 |

1. Write a Python program to draw a line using given axis values taken from a text file, with suitable label in the x axis, y axis and a title.
2. Write a Python program to plot two or more lines on same plot with suitable legends of each line.
3. Write a Python program to create multiple plots.
4. Consider the following data.

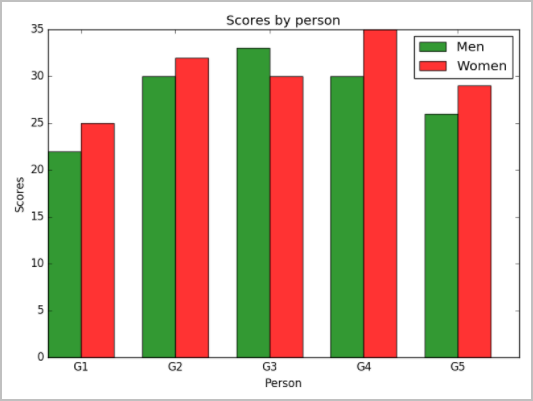
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Programming  languages: | Java | Python | PHP | JavaScript | C# | C++ |
| Popularity | 22.2 | 17.6 | 8.8 | 8 | 77 | 6.7 |

1. Write a Python programming to display a bar chart of the popularity of programming Languages.
2. Write a Python programming to display a horizontal bar chart of the popularity of programming Languages(Give Red colour to the bar chart).
3. Write a Python programming to display a bar chart of the popularity of programming Languages. Use different color for each bar.
4. Write a Python program to create bar plot of scores by group and gender. Use multiple X values on the same chart for men and women.

Sample Data:

Means (men) = (22, 30, 35, 35, 26)

Means (women) = (25, 32, 30, 35, 29)



1. Write a Python programming to create a pie chart of the popularity of programming Languages.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Programming  languages: | Java | Python | PHP | JavaScript | C# | C++ |
| Popularity | 22.2 | 17.6 | 8.8 | 8 | 77 | 6.7 |

1. Write a Python programming to create a pie chart of gold medal achievements of five most successful countries in 2016 Summer Olympics. Read the data from a csv file.

Sample data:

**medal.csv** country,gold\_medal United States,46 Great Britain,27 China,26

Russia,19 Germany,17

1. Write a Python program to draw a scatter plot comparing two subject marks of Mathematics and Science. Use marks of 10 students.

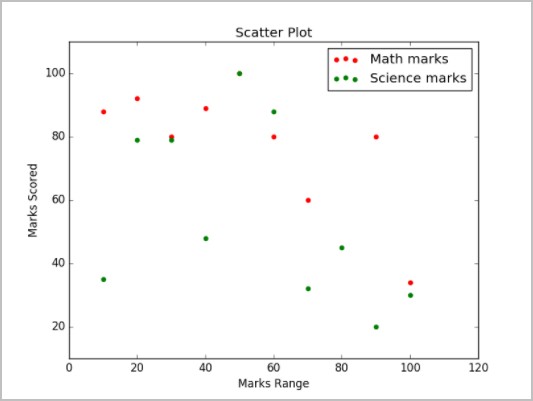
Sample data:

Test Data:

math\_marks = [88, 92, 80, 89, 100, 80, 60, 100, 80, 34]

science\_marks = [35, 79, 79, 48, 100, 88, 32, 45, 20, 30]

marks\_range = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]



# LAB CYCLE 2

# 20MCA241: DATA SCIENCE LAB

## EXERCISE 4: Introduction to Pandas CO1

* 1. Write a python program to implement List-to-Series Conversion.
  2. Write a python program to Generate the series of dates from 1st May, 2021 to 12th May, 2021 (both inclusive).
  3. Given a dictionary, convert it into corresponding dataframe and display it.
  4. Given a 2D List, convert it into corresponding dataframe and display it.
  5. Given a CSV file, read it into a dataframe and display it.
  6. Given a dataframe, sort it by multiple columns.
  7. Given a dataframe with custom indexing, convert and it to default indexing and display it.
  8. Given a dataframe, select first 2 rows and output them.
  9. Given is a dataframe showing name, occupation, salary of people. Find the average salary per occupation.
  10. Given a dataframe with NaN Values, fill the NaN values with 0.
  11. Given is a dataframe showing Company Names (cname) and corresponding Profits (profit). Convert the values of Profit column such that values in it greater than 0 are set to True and the rest are set to False.
  12. Given are 2 dataframes, with one dataframe containing Employee ID (eid), Employee Name (ename) and Stipend (stipend) and the other dataframe containing Employee ID (eid) and designation of the employee (designation). Output the Dataframe containing Employee ID (eid), Employee Name (ename), Stipend (stipend) and Position (position).

**LAB CYCLE 3**

**20MCA241: DATA SCIENCE LAB**

**EXERCISE 5 - 13**

|  |  |  |
| --- | --- | --- |
| 5 | Program to implement **k-NN classification** using any standard dataset available in the public domain and find the accuracy of the algorithm | **CO2** |
| 6 | Program to implement **Naïve Bayes Algorithm** using any standard dataset available in the public domain and find the accuracy of the algorithm | **CO2** |
| 7 | Program to implement **decision trees** using any standard dataset available in the public domain and find the accuracy of the algorithm | **CO1, CO3** |
| 8 | Program to implement linear and multiple **regression** techniques using any standard dataset available in the public domain and evaluate its performance. | **CO1, CO2** |
| 9 | Program to implement text classification using a **Support vector machine**. | **CO2, CO3** |
| 10 | Program to implement **k-means clustering** technique using any standard dataset available in the public domain | **CO2, CO3** |
| 11 | Program on a **convolutional neural network** to classify images from any the standard dataset in the public domain using the **Keras framework**. | **CO1, CO2, CO4** |
| 12 | Program to implement a simple **web crawler and scraping** web pages. | **CO1, CO2, CO5** |
| 13 | Implement problems on **natural language processing** - Part of Speech tagging, N-gram & smoothening and Chunking using **NLTK** | **CO5** |

**20MCA241: DATA SCIENCE LAB**

**Introduction to Numpy CO1**

**CYCLE - 1**

**EXERCISE 1:**

1. **Write a NumPy program to create an element-wise comparison (greater, greater\_equal,** less **and less\_equal)** of **two given** **arrays.**

**Program**

import numpy as np

x = np.array([3,5,1,2,3])

y = np.array([2,5,3,2,1])

print("Array A")

print(x)

print("\nArray B")

print(y)

print("\nA>B")

print(np.greater(x, y))

print("\nA>=B")

print(np.greater\_equal(x, y))

print("\nA<B")

print(np.less(x, y))

print("\nA<=B")

print(np.less\_equal(x, y))

**OUTPUT**

Array A

[3 5 1 2 3]

Array B

[2 5 3 2 1]

A>B

[ True False False False True]

A>=B

[ True True False True True]

A<B

[False False True False False]

A<=B

[False True True True False]

1. **Write** **a** **NumPy** **program** **to** **create** **an** **array** of **all** **the** **even** **integers from** **30 to** **70.**

**Program**

import numpy as np

x = np.arange(start=30, stop=71, step=2)

x = np.arange(30,71,2)

print(x)

**Output**

**[30 32 34 36 38 40 42 44 46 48 50 52 54 56 58 60 62 64 66 68 70]**

1. **Write a NumPy program to create a 3x3 identity** **matrix.**

**Program**

import numpy as np

x = np.identity(3)

print(x)

**Output**

**[[1. 0. 0.]**

**[0. 1. 0.]**

**[0. 0. 1.]]**

1. **Write a NumPy program to create a** vector **with values from 0 to 20 and change the sign of the numbers in the range from 9 to** **15.**

**Program**

import numpy as np

x = np.arange(21)

print("Vectors ")

print(x)

print("\nAfter changing the sign of the numbers in the range from 9 to 15:")

x[(x >= 9) & (x <= 15)] \*= -1

print(x)

**Output**

Vectors

[ 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20]

After changing the sign of the numbers in the range from 9 to 15:

[ 0 1 2 3 4 5 6 7 8 -9 -10 -11 -12 -13 -14 -15 16 17

18 19 20]

1. **Write a NumPy program** to **create a 5x5 zero matrix with elements on the main diagonal equal to 1, 2, 3, 4,** 5.

**Program**

import numpy as np

x = np.diag([1, 2, 3, 4, 5])

print(x)

**Output**

[[1 0 0 0 0]

[0 2 0 0 0]

[0 0 3 0 0]

[0 0 0 4 0]

[0 0 0 0 5]]

1. **Write a NumPy program to compute sum** of **all elements, sum** of **each column and sum** of **each row** of **a given** **array.**

**Program**

import numpy as np

x = np.array([[1,2],[0,1]])

print("Array")

print(x)

print("\nSum of all elements")

print(np.sum(x))

print("\nSum of each column")

print(np.sum(x, axis=0))

print("\nSum of each row")

print(np.sum(x, axis=1))

**Output**

Array

[[1 2]

[0 1]]

Sum of all elements

4

Sum of each column

[1 3]

Sum of each row

[3 1]

1. **Write** **a** **NumPy** **program** **to** **save** **a** **given array** **to** **a** **text** **file** **and** **load** **it.**

File: \_array.txt

# C1 C2 C3 C4

0 1 2 3

4 5 6 7

8 9 10 11

12 13 14 15

**Program**

import numpy as np

import os

x = np.arange(16).reshape(4,4)

print("Array:")

print(x)

header = 'C1 C2 C3 C4'

np.savetxt('7\_array.txt', x, fmt="%d", header=header)

print("\nAfter loading, content of the text file:")

print(np.loadtxt('7\_array.txt',dtype='i'))

**Output**

Array:

[[ 0 1 2 3]

[ 4 5 6 7]

[ 8 9 10 11]

[12 13 14 15]]

After loading, content of the text file:

[[ 0 1 2 3]

[ 4 5 6 7]

[ 8 9 10 11]

[12 13 14 15]]

1. **Write** **a** **NumPy** **program** **to** **check** **whether two** **arrays** **are** **equal** **(element** **wise)** or **not.**

**Program**

**Output**

Original arrays:

[2 2 3 2 1]

[2 3 4 3 1]

Test said two arrays are equal (element wise) or not:?

[ True False False False True]

[ True False False False True]

1. **Write a NumPy program** to **create a 4x4 array with random values, now create a new array from the said array swapping first and last** **rows.**

**Program**

import numpy as np

nums = np.arange(16, dtype='int').reshape(-1, 4)

print("Original array:")

print(nums)

print("\nNew array after swapping first and last rows of the said array:")

#new\_nums = nums[3:3:-1]

nums = nums[[-1,1,2,0]]

print(nums)

"""

num0 = list(nums[0])

num3 = list(nums[3])

nums[0] = num3

nums[3] = num0

print(nums)

"""

#####################################################################

# start through not past stop, by step

# a[start:stop:step]

# a[::-1] # all items in the array, reversed

# a[1::-1] # the first two items, reversed

# a[:-3:-1] # the last two items, reversed

# a[-3::-1] # everything except the last two items, reversed

# reference

# https://stackoverflow.com/questions/509211/understanding-slice-notation

**Output**

Original array:

[[ 0 1 2 3]

[ 4 5 6 7]

[ 8 9 10 11]

[12 13 14 15]]

New array after swapping first and last rows of the said array:

[[12 13 14 15]

[ 4 5 6 7]

[ 8 9 10 11]

[ 0 1 2 3]]

1. **Write** **a NumPy** **program** **to** **multiply** **two** **given** **arrays** **of** **same** **size** **element-by-element.**

**Program**

import numpy as np

nums1 = np.array([[2, 5, 2],[1, 5, 5]])

nums2 = np.array([[5, 3, 4],[3, 2, 5]])

print("Array1:")

print(nums1)

print("Array2:")

print(nums2)

print("\nMultiply said arrays of same size element-by-element:")

print(np.multiply(nums1, nums2))

**Output**

Array1:

[[2 5 2]

[1 5 5]]

Array2:

[[5 3 4]

[3 2 5]]

Multiply said arrays of same size element-by-element:

[[10 15 8]

[ 3 10 25]]