**2) Write code to check blank numpy array? 2 marks**

**Solution:-**

There is size attribute available for same.

>>> arr = numpy.zeros(1,0)

>>>if (arr.size==0):

print(“array is blank”)

**3) Write code to extract all even from numpy array? 2 marks**

Given numpy arry

array([ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10])

**Solution :-**

>>> arr[arr%2==0]

array([ 2, 4, 6, 8, 10])

**4) Write code to get values of specified indexes in numpy array? 2 marks**

Given array is

array([ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]) and index of(3,1,2)

**Solution:-**

>>> indexes = numpy.array([3, 1, 2 ])

>>> newarr = arr[indexes]

>>> newarr

array([4, 2, 3])

**5) Write code to get unique value counts from marks column in below data? 2 marks**

|  |  |  |
| --- | --- | --- |
|  | marks | Name |
| 0 | 98 | Shubham |
| 1 | 98 | Kapil |
| 2 | 96 | Sachin |
| 3 | 95 | Yogesh |

**Solution :-**  df.marks.unique()

**6) write code to get occurrence of value(lets say marks) in pandas dataframe when you have above dataframe? 3 marks**

**Solution :-** df.groupby('marks').size()

**7)Write code to replace all even numbers in arr with 999 without changing array ? 3 marks**

array([ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10])

**Solution:-**

>>> out = numpy.where(narray % 2 == 0, 999, narray)

>>> print(arr)

[ 1 2 3 4 5 6 7 8 9 10]

>>> print(out)

[ 1 999 3 999 5 999 7 999 9 999]

**8) Write code to get common items from numpy array ? 3 marks**

>>> x = numpy.array([11,12,13,12])

>>> y = numpy.array([17,12,10,12])

**Solution :-**

>>> numpy.intersect1d(x,y)

array([12])

**9) Write code to stack two array(output should be in desired format and student have to think about horizontal or vertically stacking)? 3 marks**

**Note :- use function of numpy library**

array a

array([[0, 1, 2, 3, 4],

[5, 6, 7, 8, 9]])

array b

array([[1, 1, 1, 1, 1],

[1, 1, 1, 1, 1]])

**Desired output :-**

array([[0, 1, 2, 3, 4],

[5, 6, 7, 8, 9],

[1, 1, 1, 1, 1],

[1, 1, 1, 1, 1]])

**Solution:-**

numpy.vstack([a, b])

**10) Write code to split array(output should be in desired format and student have to think about horizontal or vertically splitting)? 3 marks**

**Note :- use function of numpy library**

Given array :-

array([[0, 1, 2, 3, 4],

[5, 6, 7, 8, 9],

[1, 1, 1, 1, 1],

[1, 1, 1, 1, 1]])

**Desired output :-**

First array a

array([[0, 1, 2, 3, 4],

[5, 6, 7, 8, 9]])

Second array b

array([[1, 1, 1, 1, 1],

[1, 1, 1, 1, 1]])

**Solution:-**

numpy.vsplit(array,2)

**11) Write code to convert a numpy array to a dataframe of given shape? 3 marks**

ser = pd.Series(np.random.randint(1, 10, 35))

**Solution:-**

df = pd.DataFrame(ser.values.reshape(7,5))

print(df)

0 1 2 3 4

0 1 2 1 2 5

1 1 2 4 5 2

2 1 3 3 2 8

3 8 6 4 9 6

4 2 1 1 8 5

5 3 2 8 5 6

6 1 5 5 4 6

**12) Below exercise is related to pandas with matplotlib 10 marks**

**Exercise : Show Total turnover with respect to associated month in line graph**

In these exercise you will have to plot line graph with mentioned condition.

Attached data is total turnover data with specified month of different units of organization.

Generated line plot must include the following properties: –

X label name = Month

Y label name = Total turnover

**Solution :-**

import pandas as pd

import matplotlib.pyplot as plt

df = pd.read\_csv("turnover\_data.csv")

turnoverList = df ['Total\_turnover'].tolist()

months = df ['Month'].tolist()

plt.plot(months, turnoverList, label = 'Month-wise Turnover data of last year')

plt.xlabel('Month')

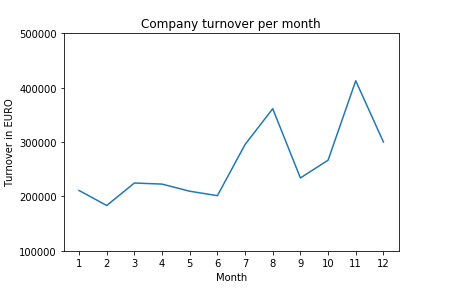
plt.ylabel('Turnover in EURO')

plt.xticks(months)

plt.title('Company turnover per month')

plt.yticks([100000, 200000, 300000, 400000, 500000])

plt.show()



**13) With above data now we will do one more exercise 10 marks**

**Exercise: Show Total turnover with respect to associated month in line graph with below specifications.**

**Specifications for plots:-**

Line Style dashed and Line-color should be green

X label name = Month Number

Y label name = Total profit number

Add a point marker.

Line width 5

**Solution :-**

import pandas as pd

import matplotlib.pyplot as plt

df = pd.read\_csv("turnover\_data.csv")

turnoverList = df ['Total\_turnover'].tolist()

months = df ['Month'].tolist()

plt.plot(months, turnoverList, label = 'Turnover data',

color='g', marker='.',

linestyle='dashed', linewidth=5)

plt.xlabel('Month')

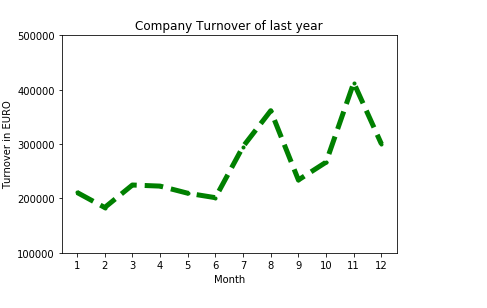
plt.ylabel('Turnover in EURO')

plt.title('Company Turnover of last year')

plt.xticks(months)

plt.yticks([100000, 200000, 300000, 400000, 500000])

plt.show()



**14)  Write program to filter words that contain atleast 2 vowels from a pandas series? 10 marks**

**Given series is:-** ser = pd.Series(['Apple', 'Orange', 'Plan', 'Python', 'Money'])

**Desired Output**

|  |  |
| --- | --- |
| 0 | Apple |
| 1 | Orange |
| 4 | Money |

**Solution :-**

from collections import Counter

mask = ser.map(lambda x: sum([Counter(x.lower()).get(i, 0) for i in list('aeiou')]) >= 2)

ser[mask]

**15) Given pandas dataframe is below 10 marks**

df = pd.DataFrame({'Z': [17,1 2, 999,13, 14, 12, 15, 999, 13, 14]})

**Task :-** calculate distance of each element from 999 and return back in desired output format.

**Desired Output :-** [1, 2, 0, 1, 2, 3, 4, 0, 1, 2].

Hint :- Use numpy functions

**Solution :-**

import pandas as pd

import numpy as np

dataframe = pd.DataFrame({'Z': [17, 12, 999, 13, 14, 12, 15, 999, 13, 14]})

izero = np.r\_[-1, (dataframe['Z'] == 999).nonzero()[0]] # indices of zeros

idx = np.arange(len(dataframe))

dataframe['Y'] = idx - izero[np.searchsorted(izero - 1, idx) - 1]

print(dataframe)

**12) Write code to read tsv file using python from these url** [**https://www.ebi.ac.uk/arrayexpress/files/E-MEXP-3682/E-MEXP-3682.sdrf.txt**](https://www.ebi.ac.uk/arrayexpress/files/E-MEXP-3682/E-MEXP-3682.sdrf.txt)

**10 marks**

**Solution :-**

import json

a1 = {}

a2 = open("Untitled.txt", "r");

for lin in a2.readlines():

filds = lin.rstrip().split("\t")

a1[filds[0]] = "\"" + filds[1]+"\" " +"\" " + filds[2]+"\" " +"\"" + filds[3]+"\" " +"\" " + filds[4]+"\" " +"\" " + filds[5]+"\" " +"\" " + filds[6]+"\" " +"\" " + filds[7]+"\" " +"\" " + filds[8]+"\" "

print(a1)

a2.close()

**1) Why numpy array uses less memory then python lists? 10 marks**