**Practical Test (Set A1) Solution**

1. Write a function to accept a list and return two list. A list of unique positive numbers and also a list of the count of the unique positive number of the input list. Example: Lst=[-1,23,-5,9,6,-23,9] would return [23,9,6] and [1,2,1]

Also copy the code in the blank space below. [5 marks]

|  |  |
| --- | --- |
| Task | Verification by Tutor |
| Correct function definition and call | 1 mark |
| Demo with input list  Lst=[-1,23,-5,9,6,-23,9] | [23, 9, 6, 9] 1 mark  [1, 3, 4, 3] 1 mark |
| Demo with input list  Lst=[-2,33,-25,7,8,33,9,7,8,33,] | [33, 7, 8, 9] 1 mark  [3, 2, 2, 1] 1 mark |

Lst=[-1,23,-5,9,6,-23,9]

def count\_pos(xlist):

idx=[]

pos\_num=[]

for c in xlist:

if c>=0:

pos\_num.append(c)

for c in pos\_num:

idx.append(xlist.index(c))

return pos\_num, idx

ans, idx=count\_pos(Lst)

1. Load the human activity data from the file ‘har\_activity1.csv’.Perform the follows tasks:

|  |  |
| --- | --- |
| Task | Verification by Tutor |
| Read in the csv file and show the first 5 rows of data and there is no missing value in the data. | 1 mark |
| Show that there is no NA, Null or missing data | 1 mark |
| Find the average age of all woman  [ 37.1680467608378 ]  Find the average age of all man  [ 39.710601719197705 ] | 1 mark  1 mark |
| Find user that height is between 1.65m to 1.80m (inclusive).  ['jose\_carlos' 'wallace'] | 1 mark |

import numpy as np

import pandas as pd

df=pd.read\_csv('har\_activity1.csv')

print(df.head())

print(df.info())

print(np.mean(df[(df.gender=='Woman')].age))

print(np.mean(df[(df.gender=='Man')].age))

print(np.unique(df[(df.how\_tall\_in\_meters>=1.65) & (df.how\_tall\_in\_meters <=1.80) ].user))

1. Based on the data in **Question 2**, plot scatter plot for y1 vs x1 for both ‘sitting’ and ‘standing’. Scale the data from 1 to 0.

[5 marks]

|  |  |
| --- | --- |
| Attribute | Verification by Tutor |
| Correct chart with point on graph for sitting in red and standing in cyan | 1 marks |
| Properly label both axis and chart title | 1 mark |
| Legend Position: lower right | 1 mark |

import matplotlib.pyplot as plt

import numpy as np

fig=plt.figure()

ax=plt.axes()

x=df[(df['class']=='sitting')].x1

y=df[(df['class']=='sitting')].y1

x=(x-np.min(x))/(np.max(x)-np.min(x))

y=(y-np.min(y))/(np.max(y)-np.min(y))

ax.plot(x,y,'or',label='sitting')

x=df[(df['class']=='standing')].x1

y=df[(df['class']=='standing')].y1

x=(x-np.min(x))/(np.max(x)-np.min(x))

y=(y-np.min(y))/(np.max(y)-np.min(y))

ax.plot(x,y,'dg',label='standing')

ax.legend(loc='lower right')

ax.axis('tight')

ax.set\_title(' y1 vs x1 for Sitting and Standing')

ax.set\_xlabel('x1')

ax.set\_ylabel('y1')

plt.show()

2 mark correct graph

