

## Numpy Assignment 2

```
import numpy as np

train_data=np.genfromtxt(r"/
train_extended.txt",delimiter=",",skip_header=1)

train_data

array([[ 1.575      ,  1.225      ,  0.375      , ...,  6.3219385 ,
        9.63883    , 10.         ],
       [ 1.2375     ,  1.         ,  0.375      , ...,  3.798833   ,
        7.654365    , 19.         ],
       [ 1.45        ,  1.1625     ,  0.4125     , ...,  7.01650125 ,
        7.257472    , 11.         ],
       ...,
       [ 1.125      ,  0.9125     ,  0.2875     , ...,  1.984465   ,
        3.118445     ,  7.         ],
       [ 1.625      ,  1.275      ,  0.4125     , ...,  7.86698625 ,
        10.489315    , 11.         ],
       [ 1.5875     ,  1.25        ,  0.3875     , ...,  7.38504475 ,
        8.788345     , 11.         ]])
```

*#What is the maximum and minimum length?*

```
maximum_length=np.max(train_data[:,0])
minimum_length=np.min(train_data[:,0])
print("maximum length is:",maximum_length)
print("minimum length is :",minimum_length)
```

maximum length is: 7.58349125

minimum length is : 0.0

*#Find out the difference between maximum and minimum length*

```
Max_min=[maximum_length-minimum_length]
print(Max_min)
```

[7.58349125]

*#Find column-wise average*

```
avg_length=np.mean(train_data[:,0])
avg_diameter=np.mean(train_data[:,1])
avg_height=np.mean(train_data[:,2])
avg_whole_weight=np.mean(train_data[:,3])
avg_weight=np.mean(train_data[:,4])
avg_shucked_weight=np.mean(train_data[:,5])
avg viscera_weight=np.mean(train_data[:,6])
avg_shell_weight=np.mean(train_data[:,7])
print(avg_length)
print(avg_diameter)
print(avg_height)
print(avg_whole_weight)
```

```

print(avg_weight)
print(avg_shucked_weight)
print(avg_viscera_weight)
print(avg_shell_weight)

1.3124480799562501
1.020320357165
0.34602787104999994
23.123436312982403
9.989370248049001
4.993180956748749
6.634229147607001
9.950615

```

```

#Find out all the age whose height is greater than 0.4.
age_greater_than_zero_point_four=train_data[train_data[:,7]>0.4]
print(age_greater_than_zero_point_four)

```

```

[[ 1.575      1.225      0.375      ...  6.3219385    9.63883
 10.         ]
 [ 1.2375     1.         0.375      ...  3.798833    7.654365
 19.         ]
 [ 1.45        1.1625     0.4125     ...  7.01650125   7.257472
 11.         ]
 ...
 [ 1.125      0.9125     0.2875     ...  1.984465    3.118445
 7.          ]
 [ 1.625      1.275      0.4125     ...  7.86698625  10.489315
 11.         ]
 [ 1.5875     1.25       0.3875     ...  7.38504475   8.788345
 11.         ]]

```

```

#What is the average height and weight of the person whose age is 10?
average_height_of_person=np.mean((train_data[:,2])
[train_data[:,7]==10])
print(average_height_of_person)
average_weight_of_person=np.mean((train_data[:,3])
[train_data[:,7]==10])
print(average_weight_of_person)

```

```

0.3833672856539947
28.243308450511314

```

```

#What is the total shell weight.
sum_of_shell_weight=np.sum([train_data[:,6]])
print(sum_of_shell_weight)

```

```

1326845.8295214002

```

```

#ow many persons belong to each and every unique age.
unique_age=np.unique([train_data[:,7]])

```

```

print(unique_age)
print("total number of unique age: ",np.count_nonzero(unique_age))

[ 1.  2.  3.  4.  5.  6.  7.  8.  9. 10. 11. 12. 13. 14. 15. 16. 17.
 18.
 19. 20. 21. 22. 23. 24. 25. 26. 27. 29.]
total number of unique age:  28

#What is the difference between shucked weight and visua weight .
difference_between_weights=train_data[:,4]-train_data[:,5]
difference_between_weights

array([5.9817445, 3.855532 , 4.1106775, ..., 2.9199985, 5.329706 ,
       9.8372765])

#What is the average height of the persons whose age is between 14 and 19.
average_height_between_ages =np.mean((train_data[:,2])
[(train_data[:,7]>14)&(train_data[:, 7]<19)])
average_height_between_ages

0.4190701892744479

#What is the average weight if we include viscous and shell weight?
total_average=np.mean(train_data[:,3]) + np.mean(train_data[:,4]) +
np.mean(train_data[:,5])
total_average

38.10598751778016

```

## Numpy Assignment 3

```
import numpy as np

abelon_data=np.genfromtxt(r"/content/Abalone -
abalone_data.csv",delimiter="," ,skip_header=1,dtype=str)
abelon_data

array([[ 'M', '91', '73', ..., '20', '30', '15'],
      [ 'M', '70', '53', ..., '10', '14', '7'],
      [ 'F', '106', '84', ..., '28', '42', '9'],
      ...,
      [ 'M', '120', '95', ..., '58', '62', '9'],
      [ 'F', '125', '97', ..., '52', '59', '10'],
      [ 'M', '142', '111', ..., '75', '99', '12']], dtype='<U3')

#1.Calculate the average length of all the seashells (Abalone)?
average_length=np.mean(abelon_data[:,1],dtype=int)
average_length

104

#2.Calculate the minimum and maximum age in the available data?
minimum_age=np.min(abelon_data[:,8].astype(int))
minimum_age

1

maximum_age=np.max(abelon_data[:,8].astype(int))
maximum_age

29

#3.Calculate the mean of meat-weight, gut weight, and dried-shell-
weight combined?
mean_of_total_weight=np.mean(abelon_data[:,5:8].astype(int))
mean_of_total_weight

51.95690687096002

#4.For all Abalones with age between [10 and 16) calculate the average
of whole weight?
avg_of_weight=np.mean(abelon_data[(abelon_data[:,8].astype(int)>=10) &
(abelon_data[:,8].astype(int)<16),4].astype(int))
avg_of_weight

214.54725274725274
```