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 .
diffrence_between_weights=train_data[:,4]-train_data[:,5]
diffrence_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
```