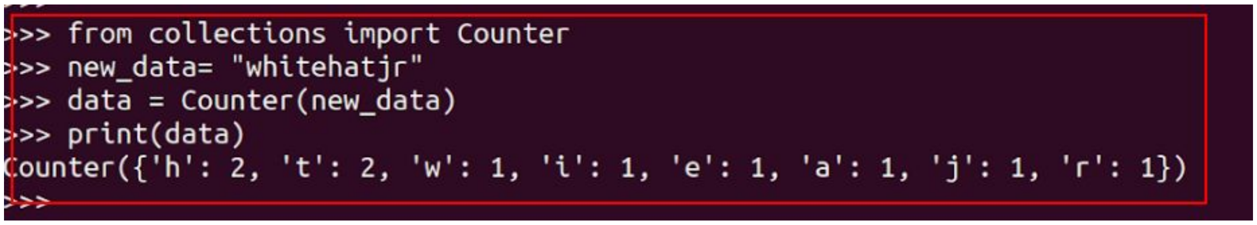
Mean,Median,Mode

What we did:
In last class we learned to visualize data by plotting multiple graphs.

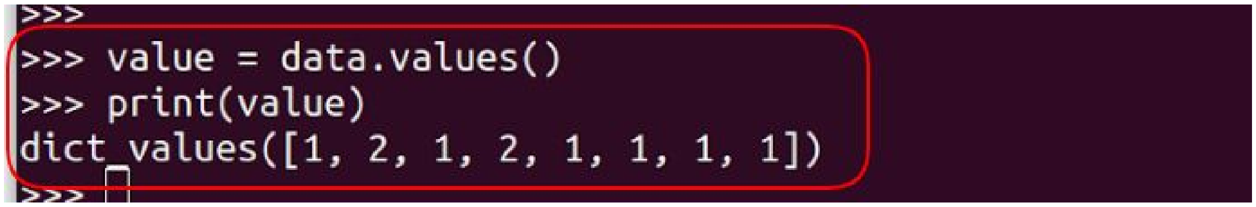
In this class we learned to find the central tendency (mean,median,mode) .

How we did it:

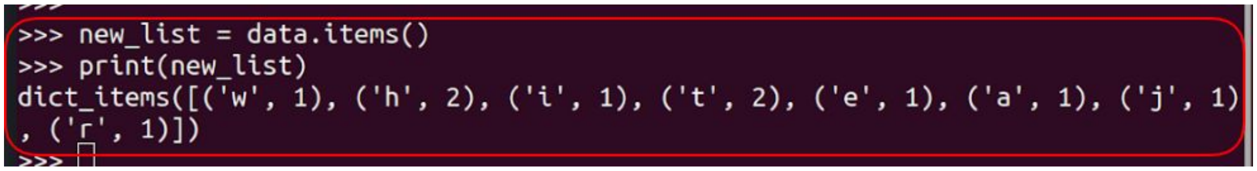
We discussed about what statistics is .
We explored some functions like Counter, items(), values()



```
>>> from collections import Counter
>>> new_data= "whitehatjr"
>>> data = Counter(new_data)
>>> print(data)
Counter({'h': 2, 't': 2, 'w': 1, 'i': 1, 'e': 1, 'a': 1, 'j': 1, 'r': 1})
>>>
```



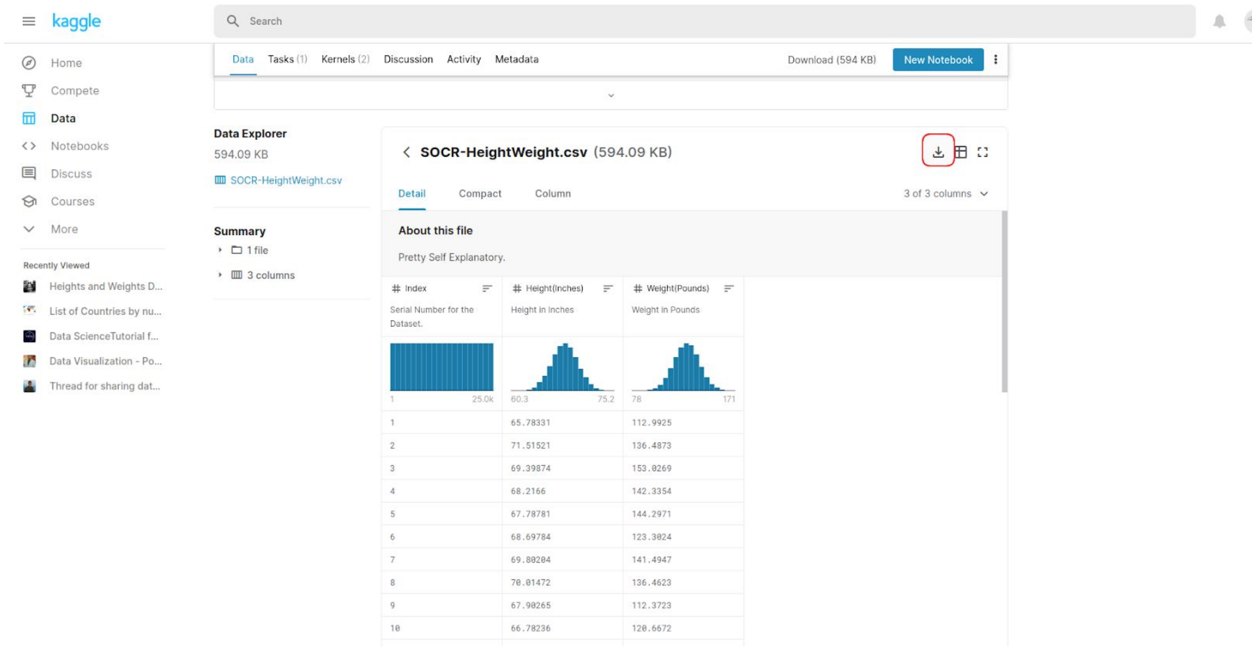
```
>>>
>>> value = data.values()
>>> print(value)
dict_values([1, 2, 1, 2, 1, 1, 1, 1])
>>>
```



```
>>> new_list = data.items()
>>> print(new_list)
dict_items([('w', 1), ('h', 2), ('i', 1), ('t', 2), ('e', 1), ('a', 1), ('j', 1)
, ('r', 1)])
>>>
```

Then we wrote code to find mean , median and mode.

We got the height and weight data from the kaggle

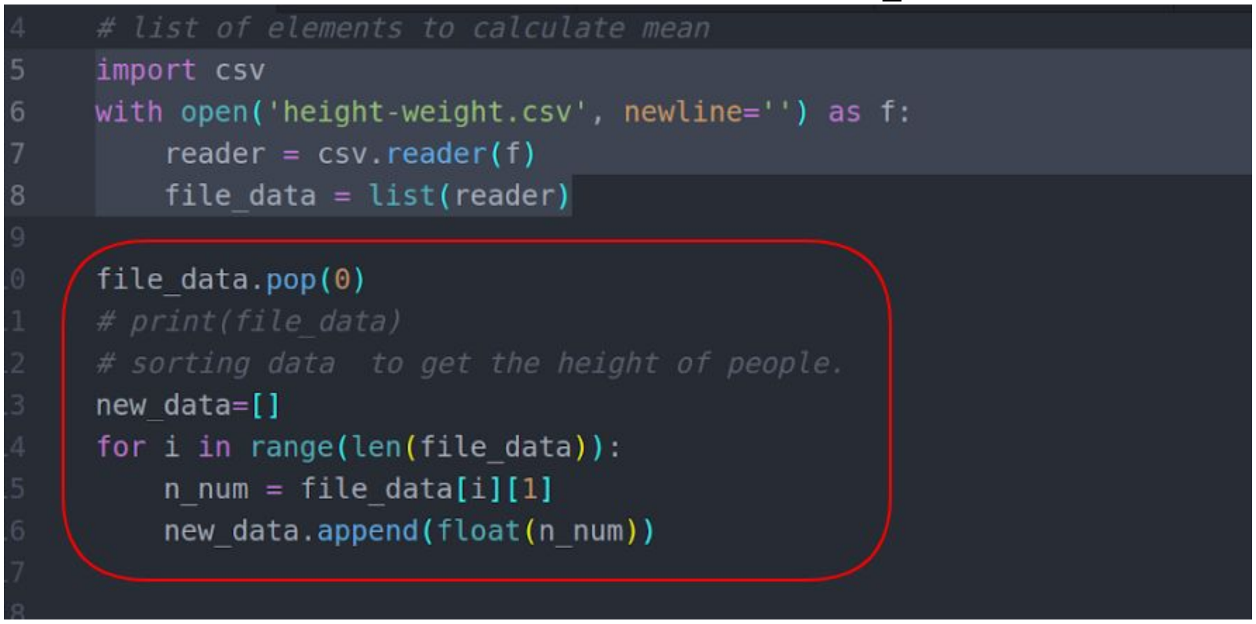


About this file

Pretty Self Explanatory.

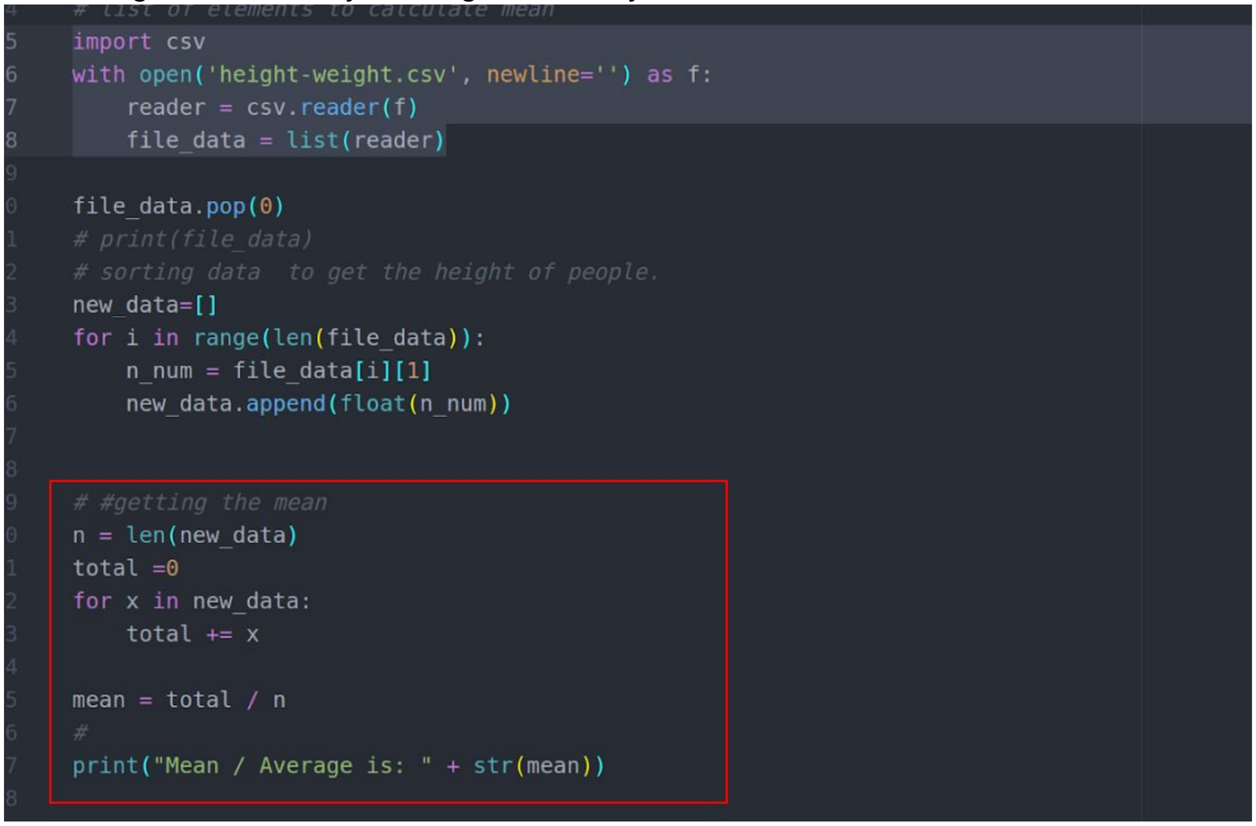
# Index	# Height(Inches)	# Weight(Pounds)
Serial Number for the Dataset.	Height in inches	Weight in Pounds
1 – 25.0k	60.3 – 75.2	78 – 171
1	65.78331	112.9925
2	71.51521	136.4873
3	69.39874	153.0269
4	68.2166	142.3354
5	67.78781	144.2971
6	68.69784	123.3024
7	69.80204	141.4947
8	70.01472	136.4623
9	67.90265	112.3723
10	66.78236	120.6672

We read the data from the csv , sorted it and stored in the new_data list.



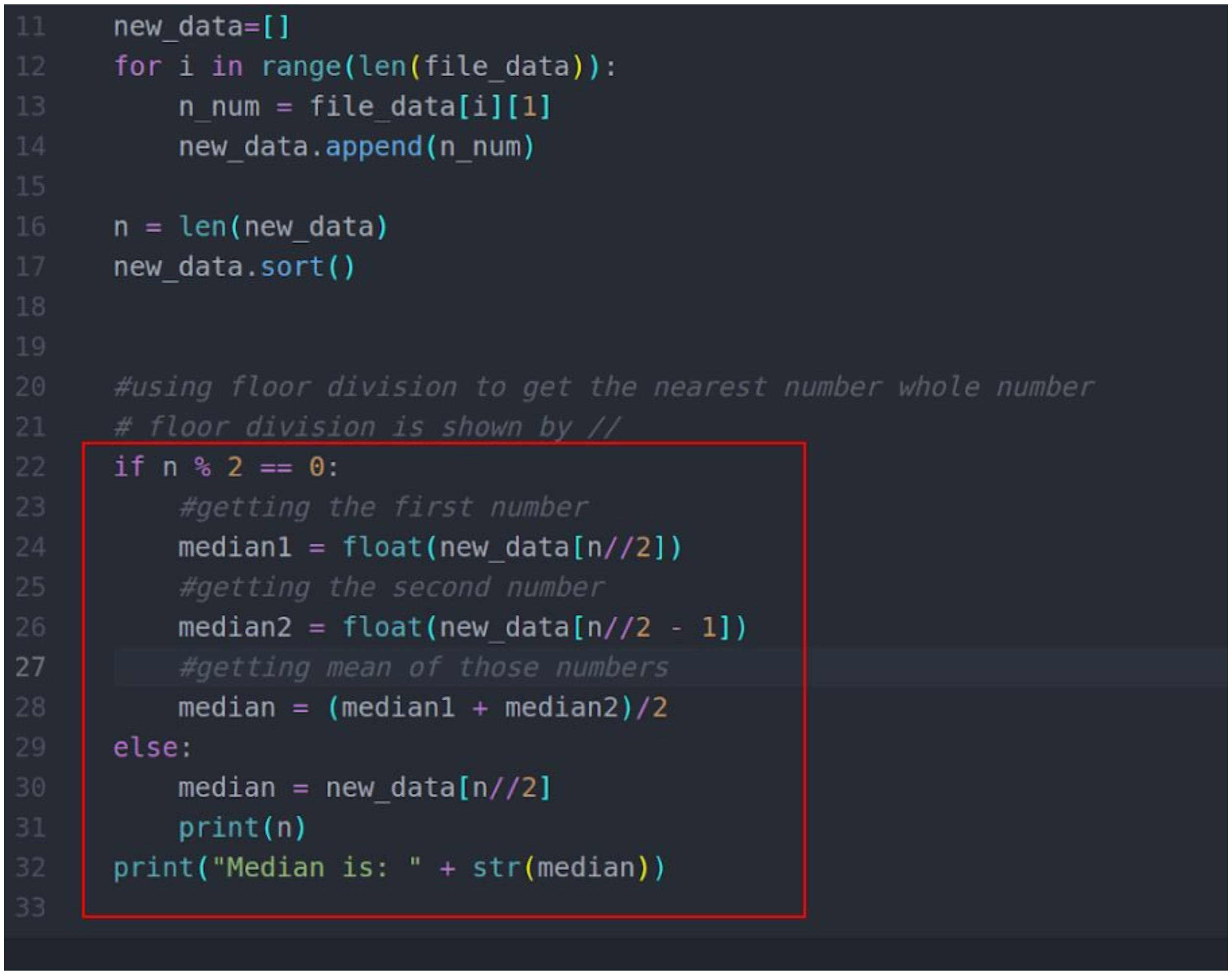
```
# list of elements to calculate mean
import csv
with open('height-weight.csv', newline='') as f:
    reader = csv.reader(f)
    file_data = list(reader)

file_data.pop(0)
# print(file_data)
# sorting data  to get the height of people.
new_data=[]
for i in range(len(file_data)):
    n_num = file_data[i][1]
    new_data.append(float(n_num))
```

Then we got the mean by dividing the total by the number of values.



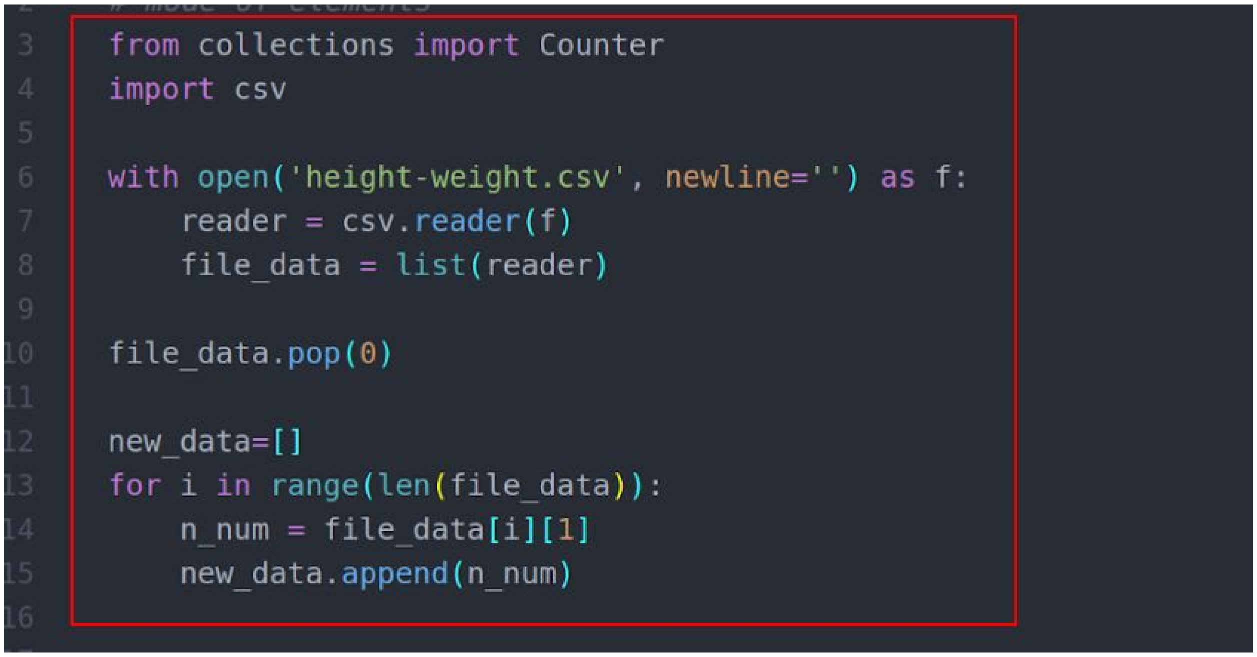
```python
# list of elements to calculate mean
import csv
with open('height-weight.csv', newline='') as f:
    reader = csv.reader(f)
    file_data = list(reader)

file_data.pop(0)
# print(file_data)
# sorting data  to get the height of people.
new_data=[]
for i in range(len(file_data)):
    n_num = file_data[i][1]
    new_data.append(float(n_num))


# #getting the mean
n = len(new_data)
total =0
for x in new_data:
    total += x

mean = total / n
#
print("Mean / Average is: " + str(mean))
```

Then we find the median

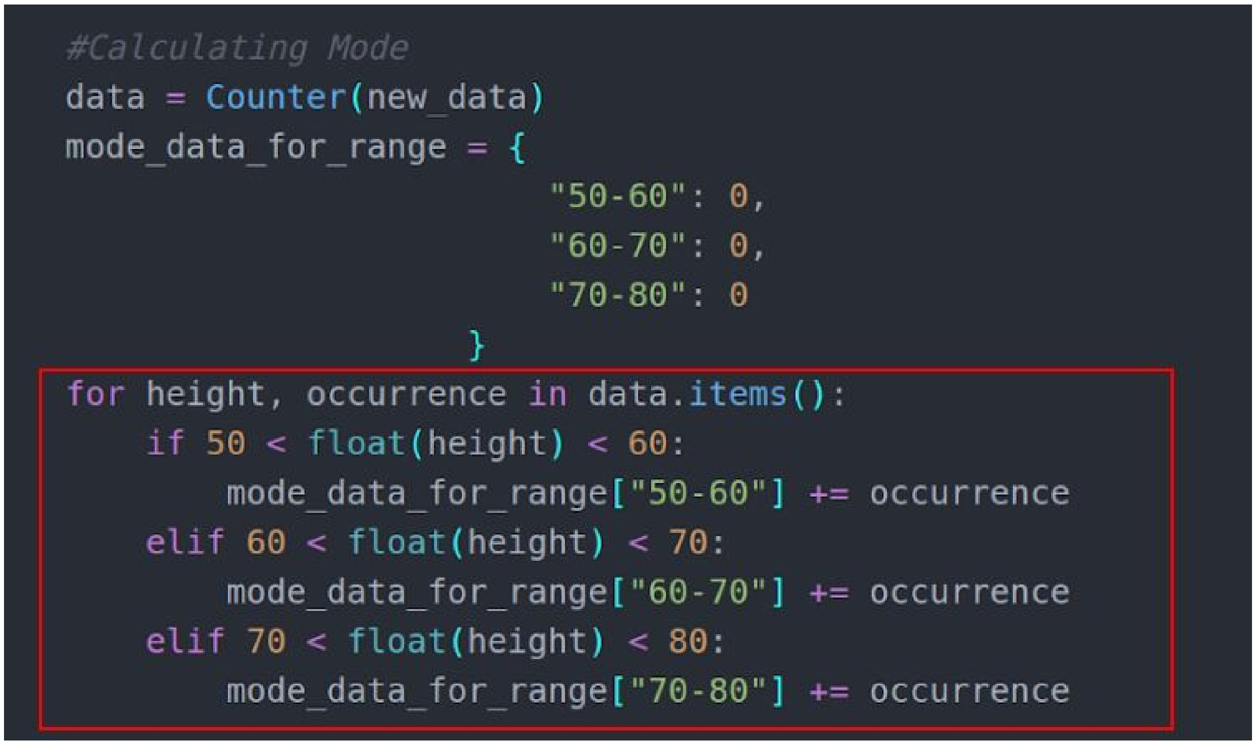


```
new_data=[]
for i in range(len(file_data)):
    n_num = file_data[i][1]
    new_data.append(n_num)

n = len(new_data)
new_data.sort()


#using floor division to get the nearest number whole number
# floor division is shown by //
if n % 2 == 0:
    #getting the first number
    median1 = float(new_data[n//2])
    #getting the second number
    median2 = float(new_data[n//2 - 1])
    #getting mean of those numbers
    median = (median1 + median2)/2
else:
    median = new_data[n//2]
    print(n)
print("Median is: " + str(median))
```

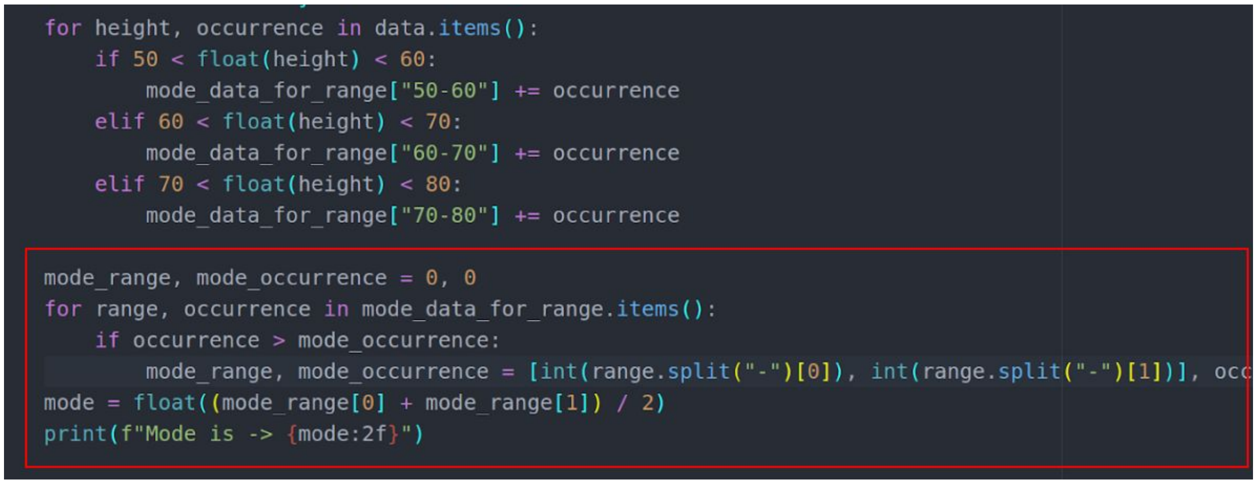
We found the mode



```
from collections import Counter
import csv

with open('height-weight.csv', newline='') as f:
    reader = csv.reader(f)
    file_data = list(reader)

file_data.pop(0)

new_data=[]
for i in range(len(file_data)):
    n_num = file_data[i][1]
    new_data.append(n_num)
```



```
#Calculating Mode
data = Counter(new_data)
mode_data_for_range = {
                        "50-60": 0,
                        "60-70": 0,
                        "70-80": 0
                    }
for height, occurrence in data.items():
    if 50 < float(height) < 60:
        mode_data_for_range["50-60"] += occurrence
    elif 60 < float(height) < 70:
        mode_data_for_range["60-70"] += occurrence
    elif 70 < float(height) < 80:
        mode_data_for_range["70-80"] += occurrence
```



```
for height, occurrence in data.items():
    if 50 < float(height) < 60:
        mode_data_for_range["50-60"] += occurrence
    elif 60 < float(height) < 70:
        mode_data_for_range["60-70"] += occurrence
    elif 70 < float(height) < 80:
        mode_data_for_range["70-80"] += occurrence

mode_range, mode_occurrence = 0, 0
for range, occurrence in mode_data_for_range.items():
    if occurrence > mode_occurrence:
        mode_range, mode_occurrence = [int(range.split("-")[0]), int(range.split("-")[1])], occ
mode = float((mode_range[0] + mode_range[1]) / 2)
print(f"Mode is -> {mode:2f}")
```

What’s next?

In the next class, we will learn about descriptive statistics.