In [*]:
```
import numpy as np
import pandas as pd
import scipy as sp
```

In [2]:
```
%matplotlib inline
import matplotlib.pyplot as plt
plt.style.use('ggplot')
```

In [3]:
```
%%file hw_data.csv
id,sex,weight,height
1,M,190,77
2,F,120,70
3,F,110,68
4,M,150,72
5,O,120,66
6,M,120,60
7,F,140,70
```

```
Overwriting hw_data.csv
```

Python

1. Finish creating the following function that takes a list and returns the average value.

Add each element in the list to `total` and return `total`

DO NOT use a library function nor `sum()`

In [4]:
```
def average(my_list):
    total = 0
    for item in my_list:
         total = total + item
    total = total / len(my_list)
    return total

average([1,2,1,4,3,2,5,9])
```

Out[4]:
```
3.375
```

2. Using a Dictionary keep track of the count of numbers (or items) from a list

In [5]:
```
def counts(my_list):
    counts = dict()
    for item in my_list:
        if item in counts:
            counts[item] +=1
        else:
            counts[item] = 1
    return counts

counts([1,2,1,4,3,2,5,9])
```

Out[5]:
```
{1: 2, 2: 2, 4: 1, 3: 1, 5: 1, 9: 1}
```

3. Using the `counts()` function you created above and the `.split()` function, return a dictionary of most occuring words from the following paragraph. Bonus, remove punctuation from words.

In [6]:
```
paragraph_text = '''
For a minute or two she stood looking at the house, and wondering what
The Fish-Footman began by producing from under his arm a great letter,
Then they both bowed low, and their curls got entangled together.
Alice laughed so much at this, that she had to run back into the wood
Alice went timidly up to the door, and knocked.
‘There’s no sort of use in knocking,’ said the Footman, ‘and that for
‘Please, then,’ said Alice, ‘how am I to get in?’
‘There might be some sense in your knocking,’ the Footman went on with
‘I shall sit here,’ the Footman remarked, ‘till tomorrow—’
At this moment the door of the house opened, and a large plate came sk

import re

paragraph_text_np = re.sub(r'[^\w\s]',' ',paragraph_text) # Doesn't wo
                                                            # Technicall
counts(paragraph_text_np.split())
```

Out[6]:
```
{'For': 3,
 'a': 16,
 'minute': 1,
 'or': 2,
 'two': 2,
 'she': 7,
 'stood': 1,
 'looking': 2,
 'at': 6,
 'the': 32,
 'house': 2,
 'and': 18,
 'wondering': 1,
 'what': 2,
 'to': 15,
 'do': 1,
 'next': 2,
 'when': 2,
 'suddenly': 1,
```

4. Read in a file using `open()` and iterated through the file line-by-line write each line from the file to a new file in a `title()`-ized. Create your own file for input

```
This is the first line -> This Is The First Line
```

Hint: There's a function to do this

In [7]:
```


lines = ["test file", 'This is the first line', 'This is the second li

with open('test_file.txt', "w") as test_file: # Making a random file f
    test_file.write('\n'.join(lines))

with open("test_file.txt") as test_read: # I am sure there is a more e
         print (test_read.read())        # I couldn't figure it out


with open('title_test_file.txt',"w") as title_test_file:
        with open('test_file.txt', "r") as test_file:
            for line in test_file:
                 line = line.title()
                 title_test_file.write(line)

with open("title_test_file.txt") as test_read:
        print (test_read.read())
```

```
test file
This is the first line
This is the second line
This is the third line
Test File
This Is The First Line
This Is The Second Line
This Is The Third Line
```

Numpy

1. Given a list, find the average using a numpy function.

In [8]:
```
simple_list = [1,2,1,4,3,2,5,9]

np.mean(simple_list)
```
Out[8]:
```
3.375
```

2. Given two lists of Heights and Weights of individual, calculate the BMI of those individuals, without writing a `for-loop`

In [9]:
```
heights = [174, 173, 173, 175, 171]
weights = [88, 83, 92, 74, 77]

bmi = weights / (np.square(heights))

bmi
```
Out[9]:
```
array([0.00290659, 0.00277323, 0.00307394, 0.00241633, 0.00263329])
```

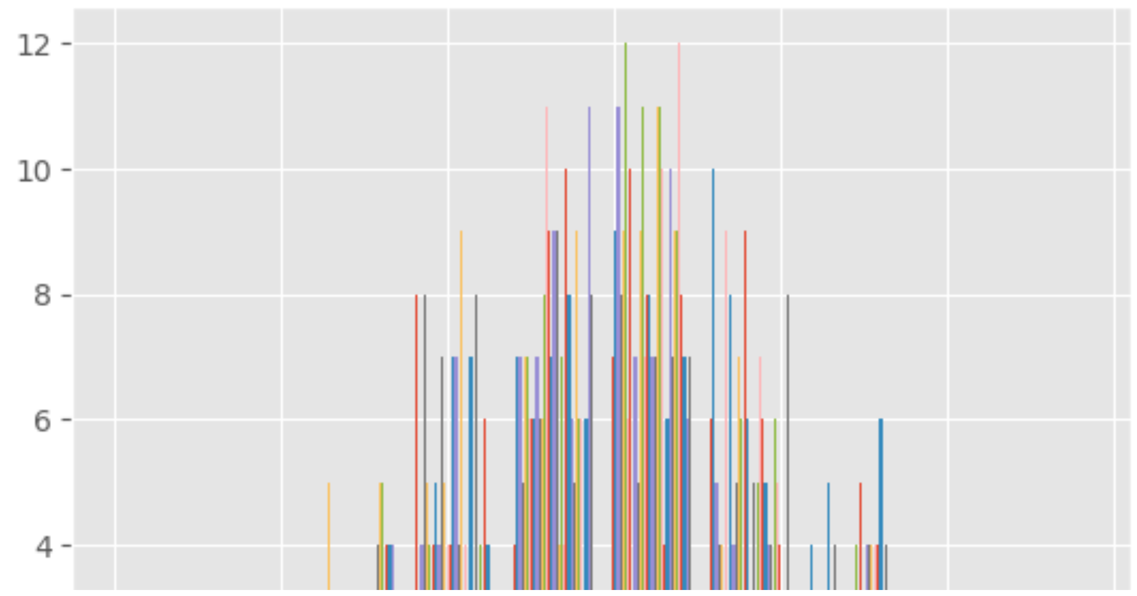
3. Create an array of length 20 filled with random values (between 0 to 1)

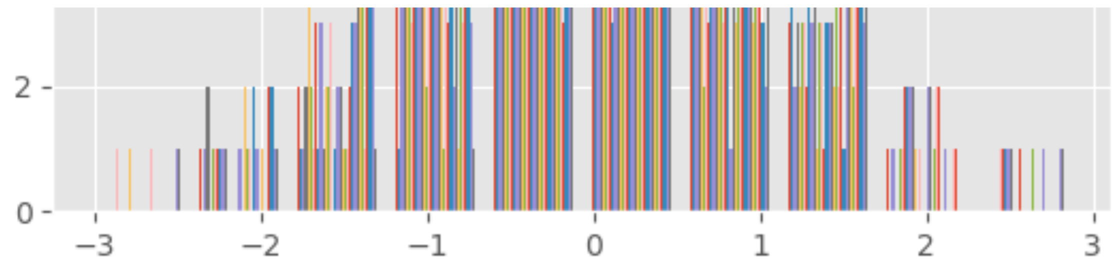
In [10]:
```
np.random.rand(1,20)
```
Out[10]:
```
array([[0.712942  , 0.87984612, 0.30754256, 0.03092126, 0.35339384,
        0.01661483, 0.45555996, 0.92041275, 0.25654438, 0.92911781,
        0.38540922, 0.31399911, 0.1200029 , 0.73049234, 0.29165696,
        0.78691093, 0.80444715, 0.82512303, 0.03004801, 0.13327666]])
```

4. Create an array with at least 1000 random numbers from normal distributions (normal). Then, plot a histogram of these values (`plt.hist`).

In [11]:
```
rando = np.random.randn(32,32) #1024 numbers

plt.hist(rando)
```
Out[11]:
```
(array([[ 0.,  1.,  2.,  8.,  4.,  7.,  6.,  3.,  1.,  0.],
        [ 0.,  0.,  1.,  1.,  7.,  9., 10.,  4.,  0.,  0.],
        [ 0.,  1.,  1.,  4.,  7., 11.,  5.,  2.,  1.,  0.],
        [ 0.,  2.,  2.,  8.,  5.,  8.,  4.,  3.,  0.,  0.],
        [ 0.,  0.,  5.,  5.,  7.,  9.,  4.,  2.,  0.,  0.],
        [ 0.,  1.,  2.,  4.,  7., 12.,  2.,  3.,  1.,  0.],
        [ 1.,  1.,  1.,  3.,  6.,  6.,  9.,  3.,  1.,  1.],
        [ 0.,  1.,  3.,  4.,  6., 10.,  3.,  2.,  2.,  1.],
        [ 0.,  1.,  1.,  5.,  6.,  3.,  8.,  5.,  2.,  1.],
        [ 0.,  1.,  3.,  4.,  7.,  7.,  4.,  3.,  2.,  1.],
        [ 0.,  1.,  1.,  7.,  6.,  5.,  5.,  4.,  2.,  1.],
        [ 1.,  0.,  2.,  5.,  6.,  9.,  7.,  1.,  1.,  0.],
        [ 0.,  0.,  2.,  2.,  8., 11.,  6.,  3.,  0.,  0.],
        [ 0.,  0.,  3.,  4., 11.,  7.,  3.,  3.,  1.,  0.],
        [ 0.,  0.,  0.,  4.,  9.,  8.,  9.,  1.,  0.,  1.],
        [ 0.,  0.,  1.,  7.,  7.,  8.,  6.,  3.,  0.,  0.],
        [ 0.,  1.,  2.,  7.,  9.,  7.,  1.,  3.,  2.,  0.],
        [ 0.,  0.,  2.,  4.,  9.,  7.,  5.,  3.,  2.,  0.],
        [ 0.,  2.,  1.,  9.,  4., 11.,  3.,  2.,  0.,  0.],
        [ 0.,  1.,  1.,  1.,  7., 11.,  5.,  4.,  1.,  1.],
        [ 1.,  1.,  1.,  4.,  4., 10.,  7.,  2.,  2.,  0.],
        [ 0.,  0.,  2.,  3., 10.,  4.,  6.,  5.,  2.,  0.],
        [ 0.,  2.,  3.,  7.,  8.,  6.,  5.,  1.,  0.,  0.],
        [ 0.,  1.,  3.,  2.,  6., 10.,  4.,  4.,  1.,  1.],
        [ 0.,  0.,  4.,  8.,  5.,  7.,  4.,  4.,  0.,  0.],
        [ 0.,  1.,  5.,  1.,  9.,  9.,  3.,  4.,  0.,  0.],
        [ 0.,  0.,  5.,  4.,  6.,  9.,  6.,  2.,  0.,  0.],
        [ 0.,  0.,  1.,  3.,  6., 12.,  5.,  4.,  1.,  0.],
        [ 0.,  2.,  4.,  6.,  3.,  8.,  4.,  4.,  1.,  0.],
        [ 0.,  2.,  4.,  4.,  6.,  7.,  3.,  6.,  0.,  0.],
        [ 1.,  1.,  4.,  3., 11.,  6.,  2.,  3.,  0.,  1.],
        [ 1.,  1.,  1.,  1.,  8.,  7.,  8.,  4.,  0.,  1.]]),
 array([-3.0158095 , -2.42580191, -1.83579431, -1.24578672, -0.655779
13,
        -0.06577153,  0.52423606,  1.11424365,  1.70425125,  2.294258
84,
         2.88426643]),
 <a list of 32 BarContainer objects>)
```





Pandas

1. Read in a CSV () and display all the columns and their respective data types

In [12]:
```
df = pd.read_csv('hw_data.csv')
df.columns
```

Out[12]:
```
Index(['id', 'sex', 'weight', 'height'], dtype='object')
```

2. Find the average weight

In [13]:
```
np.mean(df['weight']) # 135.714286
```

Out[13]:
```
135.71428571428572
```

3. Find the Value Counts on column `sex`

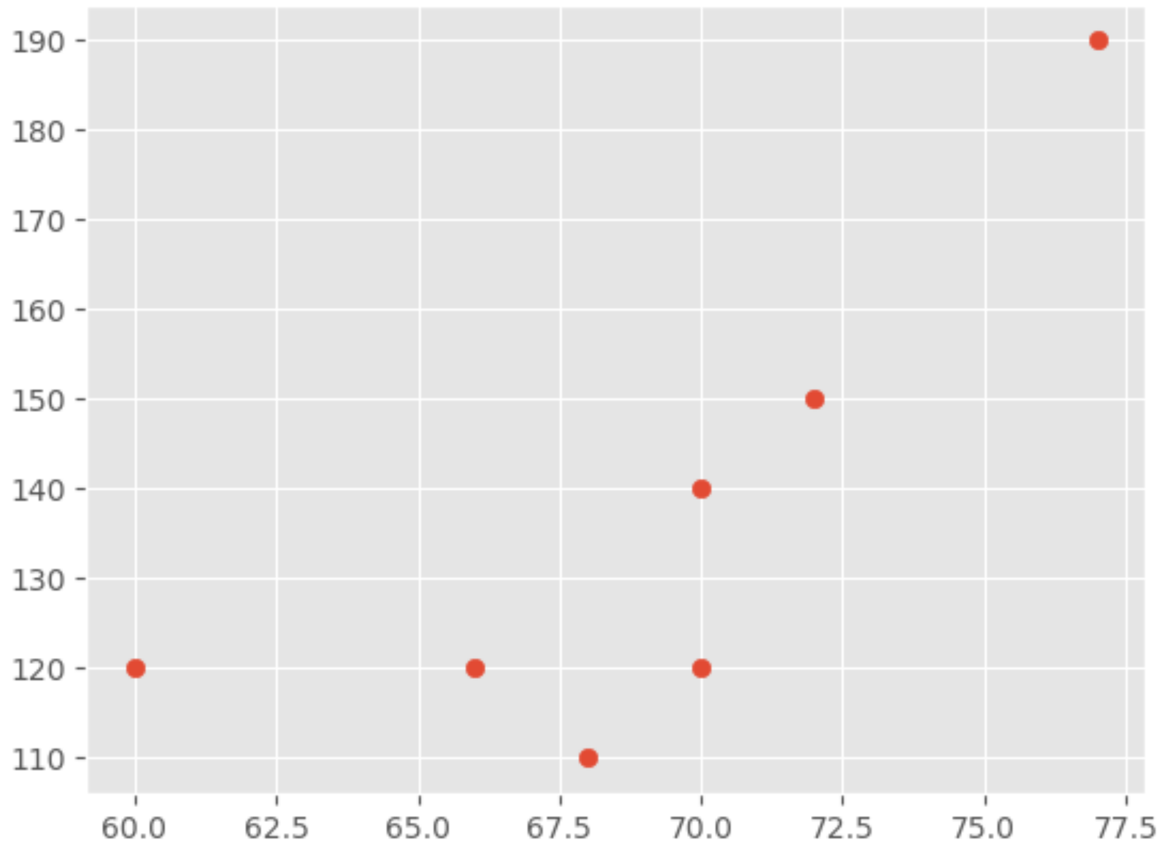
In [14]:
```
counts(df['sex']) # {'M': 3, 'F': 3, 'O': 1}
```

Out[14]:
```
{'M': 3, 'F': 3, 'O': 1}
```

4. Plot Height vs. Weight

In [15]:
```
plt.scatter(df['height'],df['weight'])
```

Out[15]:
```
<matplotlib.collections.PathCollection at 0x7f7e2cf930a0>
```



5. Calculate BMI and save as a new column

In [16]:
```
bmi = df['weight'] / (np.square(df['height']))

df['BMI'] = bmi

df
```

Out[16]:

	id	sex	weight	height	BMI
0	1	M	190	77	0.032046
1	2	F	120	70	0.024490
2	3	F	110	68	0.023789
3	4	M	150	72	0.028935
4	5	O	120	66	0.027548
5	6	M	120	60	0.033333
6	7	F	140	70	0.028571

6. Save sheet as a new CSV file `hw_dataB.csv`

In [17]:
```
df.to_csv("hw_dataB.csv")
```

Run the following (Mac)

In [18]:
```
!cat hw_dataB.csv
```

```
,id,sex,weight,height,BMI
0,1,M,190,77,0.03204587620172036
1,2,F,120,70,0.024489795918367346
2,3,F,110,68,0.02378892733564014
3,4,M,150,72,0.028935185185185185
4,5,O,120,66,0.027548209366391185
5,6,M,120,60,0.03333333333333333
6,7,F,140,70,0.02857142857142857
```

Run the following (Windows)

In [19]:
```
#!type hw_dataB.csv Mac user here
```