In [107…

```
import numpy as np
import pandas as pd
import scipy as sp
import os
```

In [2]:

```
%matplotlib inline
import matplotlib.pyplot as plt
plt.style.use('ggplot')
```

In [ ]:

```
#### reversed height and weight because I think they were backwards
```

In [127…

```
%%file hw_data.csv
id,sex,height,weight
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 [5]:

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

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

Out[5]:

```
3.375
```

In [6]:

```
# checking math
np.mean([1,2,1,4,3,2,5,9])
```

Out[6]:

```
3.375
```

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

In [19]:

```
def counts(my_list):
    counts = dict()
    for item in my_list:
        if item not in counts.keys():
            counts[item] = 1
        else:
            counts[item] = counts[item] + 1
    return counts

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

Out[19]:

```
{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 [83]:

```
paragraph_text = '''
For a minute or two she stood looking at the house, and wondering what to do next, when suddenly a footman in livery came running
The Fish-Footman began by producing from under his arm a great letter, nearly as large as himself, and this he handed over to the
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 for fear of their hearing her; and when she next peeped out
Alice went timidly up to the door, and knocked.
‘There’s no sort of use in knocking,’ said the Footman, ‘and that for two reasons. First, because I’m on the same side of the door
‘Please, then,’ said Alice, ‘how am I to get in?’
‘There might be some sense in your knocking,’ the Footman went on without attending to her, ‘if we had the door between us. For in
‘I shall sit here,’ the Footman remarked, ‘till tomorrow—’
At this moment the door of the house opened, and a large plate came skimming out, straight at the Footman’s head: it just grazed h

punc = '''!;:‘'"'\,./?&_~—\n'''

for word in paragraph_text:
    if word in punc:
        paragraph_text = paragraph_text.replace(word, "")

paragraph_count = counts(paragraph_text.split(" "))
max_value = max(paragraph_count.values())

value = {i for i in paragraph_count if paragraph_count[i]== max_value}
print("the most common word is:", value, "with", max_value, "instances")
```

```
the most common word is: {'the'} with 32 instances
```

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 [71]:

```
raven = open("raven.txt", "r")
ravenLines = raven.readlines()

newFile = open('newRaven.txt','w')

for line in ravenLines:
    newFile.write(line.title())

raven.close()
newFile.close()

# check for success
newRaven = open("newRaven.txt", "r")
newRaven.readlines()
```

Out[71]:

```
['The Raven\n',
 'By Edgar Allan Poe\n',
 'Once Upon A Midnight Dreary, While I Pondered, Weak And Weary,\n',
 'Over Many A Quaint And Curious Volume Of Forgotten Lore\n',
 'While I Nodded, Nearly Napping, Suddenly There Came A Tapping,\n',
 'As Of Some One Gently Rapping, Rapping At My Chamber Door.\n',
 'Tis Some Visitor, I Muttered, Tapping At My Chamber Door\n',
 'Only This And Nothing More.']
```

Numpy

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

In [84]:

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

print(np.mean(simple_list))
```

```
3.375
```

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

In [94]:

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

bmi = np.array(weights) / ((np.array(heights)) / 100) **2
print(bmi)
```

```
[29.06592681 27.73229978 30.73941662 24.16326531 26.33288875]
```

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

In [93]:

```
np.random.rand(20,)
```

Out[93]:

```
array([0.84018402, 0.52260151, 0.76305389, 0.14597018, 0.6916868 ,
       0.364418  , 0.76510702, 0.98617253, 0.08609056, 0.20197657,
       0.29077918, 0.33758239, 0.0573407 , 0.92973008, 0.27779317,
       0.71712866, 0.59669654, 0.35985709, 0.96590839, 0.67489956])
```

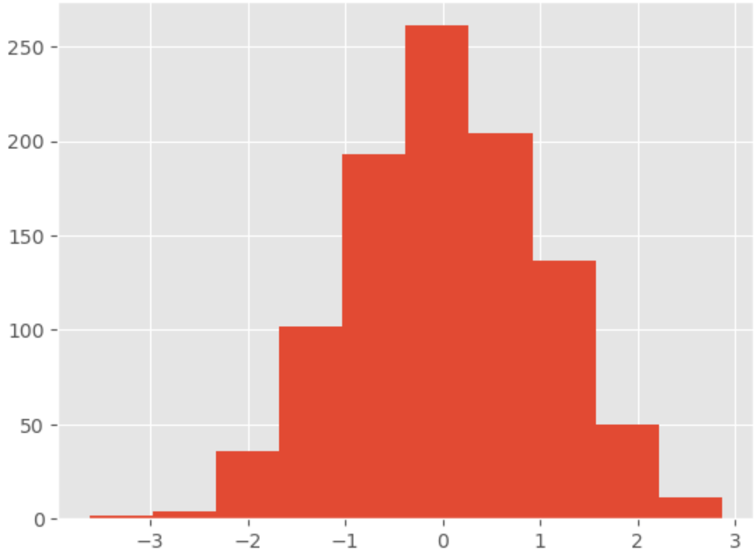
4. Create an array with at least 1000 random numbers from normal distributions (normal). Then, plot a histogram of these values (`plt.hist`).

In [100…

```
plt.hist(np.random.randn(1000,))
```

Out[100]:

```
(array([  2.,   4.,  36., 102., 193., 261., 204., 137.,  50.,  11.]),
 array([-3.61954011, -2.97146615, -2.32339219, -1.67531823, -1.02724427,
        -0.3791703 ,  0.26890366,  0.91697762,  1.56505158,  2.21312554,
         2.8611995 ]),
 <BarContainer object of 10 artists>)
```



Pandas

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

In [128…

```
bmi_df = pd.read_csv("hw_data.csv")
bmi_df = pd.DataFrame(bmi_df)
print(bmi_df)
bmi_df.dtypes
```

```
   id sex  height  weight
0   1   M     190      77
1   2   F     120      70
2   3   F     110      68
3   4   M     150      72
4   5   O     120      66
5   6   M     120      60
6   7   F     140      70
```

Out[128]:

```
id         int64
sex       object
height     int64
weight     int64
dtype: object
```

2. Find the average weight

In [129…

```
np.mean(bmi_df["weight"])
```

Out[129]:

```
69.0
```

3. Find the Value Counts on column `sex`

In [130…

```
bmi_df["sex"].value_counts()
```

Out[130]:

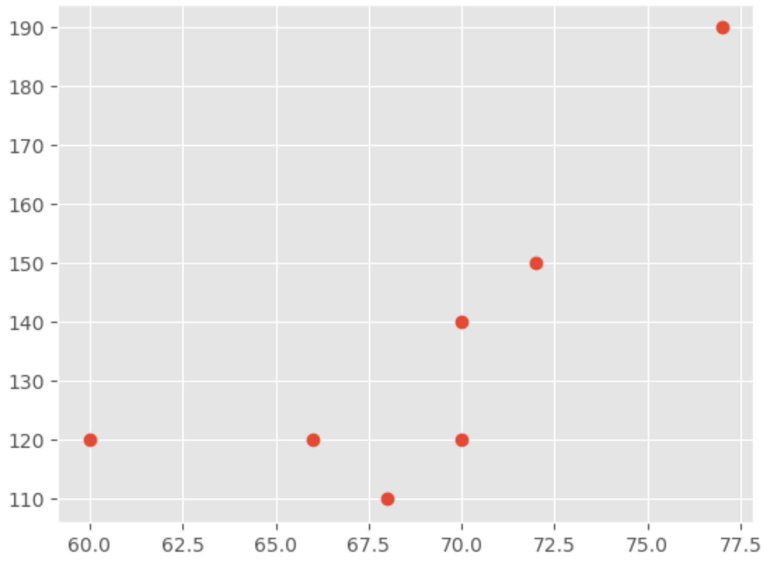
```
sex
M    3
F    3
O    1
Name: count, dtype: int64
```

4. Plot Height vs. Weight

In [131…

```
x = bmi_df["weight"]
y = bmi_df["height"]

plt.plot(x,y, "o")
plt.show()
```



5. Calculate BMI and save as a new column

In [135…

```
bmi_df["bmi"] = round(bmi_df["weight"]/((bmi_df["height"]/100)**2), 2)
print(bmi_df)
```

```
   id sex  height  weight    bmi
0   1   M     190      77  21.33
1   2   F     120      70  48.61
2   3   F     110      68  56.20
3   4   M     150      72  32.00
4   5   O     120      66  45.83
5   6   M     120      60  41.67
6   7   F     140      70  35.71
```

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

In [136…

```
bmi_df.to_csv("hw_dataB.csv")
```

Run the following (Mac)

In [ ]:

```
!cat hw_dataB.csv
```

Run the following (Windows)

In [137…

```
!type hw_dataB.csv
```

```
,id,sex,height,weight,bmi
0,1,M,190,77,21.33
1,2,F,120,70,48.61
2,3,F,110,68,56.2
3,4,M,150,72,32.0
4,5,O,120,66,45.83
5,6,M,120,60,41.67
6,7,F,140,70,35.71
```

In [ ]: