In [2]:
```
import numpy as np
import pandas as pd
import scipy as sp
```

In [3]:
```
%matplotlib inline
import matplotlib.pyplot as plt
plt.style.use('ggplot')
```

In [4]:
```
%%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 [33]:
```
my_list = [5,15,22,74,11,131,6,8,2]

total = 5 + 15 + 22 + 74 + 11 + 131 + 6 + 8 + 2
print("The total of my list is " + str(total) + ".")

count = len(my_list)
print("The number of values in my list is " + str(count) + ".")

avg_val = total / count
print("The average value of an item in my list is " + str(avg_val) + "
```

```
The total of my list is 274.
The number of values in my list is 9.
The average value of an item in my list is 30.444444444444443.
```

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

In [31]:
```
def counts(my_list):
    counts = dict()
    for item in my_list:
        if item in counts:
            counts[item] += 1
        else:
            counts[item] = 1
    return counts

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

```
{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 [39]:
```
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

#Ummmm... this was quite difficult! In the end, I follow though.

#Replace the punctuation with a blank space- I'm tracking so far
no_punc = paragraph_text
for chararcter in '?!;:,.'—'()-':
    no_punc = no_punc.replace(chararcter, ' ')

#title case
no_punc_title = no_punc.title()

#prepare for big dict
big_dict = no_punc_title.split()

#whip it out
counts(big_dict)
```

Out[39]:
```
{'For': 6,
 'A': 16,
 'Minute': 1,
 'Or': 2,
 'Two': 2,
 'She': 8,
 'Stood': 1,
 'Looking': 2,
 'At': 7,
 'The': 34,
 'House': 2,
 'And': 19,
 '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 [20]:
```
#nogo on the /n here? I don't understand what is being asked about lin
L = ["This is the first line.", "This is the second line.", "This is t

#writelines()
file1 = open('try2.txt', 'w')
file1.writelines(L)
file1.close()

#readlines()
file1 = open('try2.txt', 'r')
Lines = file1.readlines()

#something
count = 0
for line in Lines:
    count += 1
    print("Line{}: {}".format(count, line.strip()))
```

```
Line1: This is the first line.This is the second line.This is the thi
rd line.
```

Numpy

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

In [32]:
```
simple_list = [1,2,1,4,3,2,5,9]

np.average(simple_list)
```

Out[32]:
```
3.375
```

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

In [25]:
```
heights = [174, 173, 173, 175, 171]
weights = [88, 83, 92, 74, 77]

meters = np.divide(heights, 100)
meters_sq = np.multiply(meters, meters)
BMI = np.divide(weights, meters_sq)
print("The BMI is: " + str(BMI))
```

```
The BMI is: [29.06592681 27.73229978 30.73941662 24.16326531 26.33288
875]
```

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

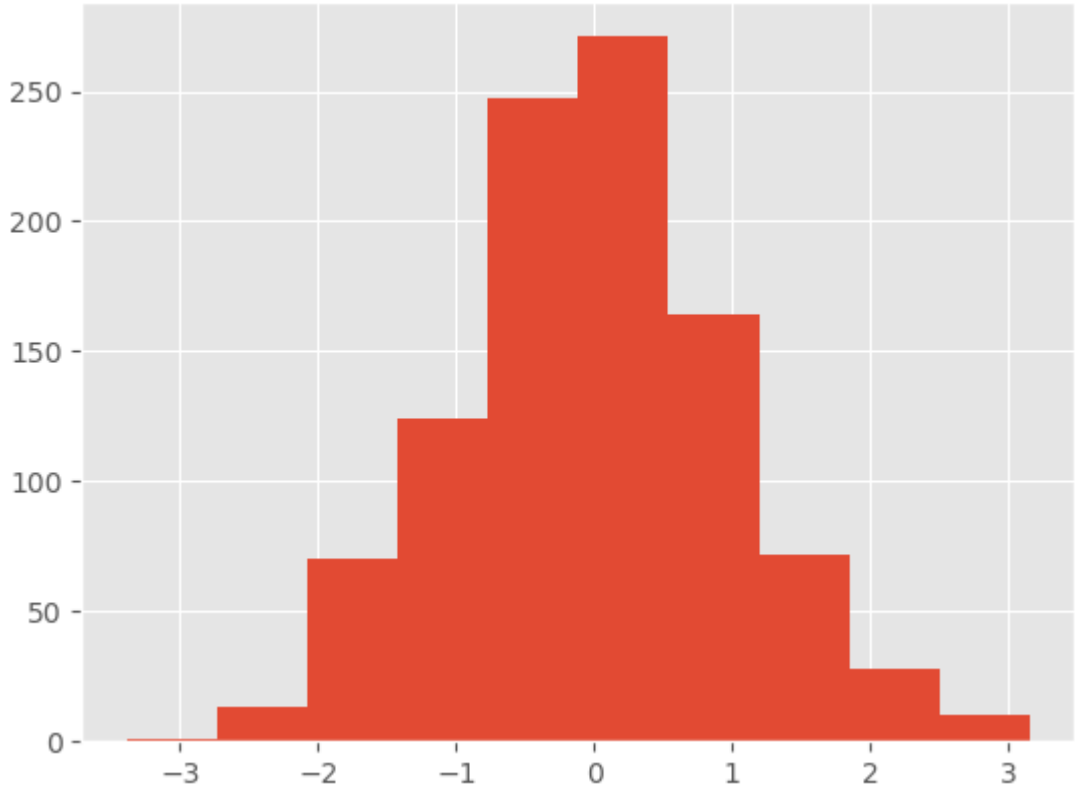
In [27]:
```
mant_array = np.random.randn(1, 20)
mant_array
```

Out[27]:
```
array([[ 0.59332193, -0.70790471,  1.67796938, -0.46180915,  0.754871
34,
         0.65131997, -0.74424887, -1.5655081 , -0.96782567, -0.525190
82,
        -0.7566715 ,  1.04717994, -0.0997502 , -0.44959152, -0.427447
16,
         0.64044947, -1.46518485,  0.72337746, -0.44495547,  0.122327
16]])
```

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

In [28]:
```
# Array with 1000 random numbers
array_bnnights = np.random.normal(size=1000)

# Histogram
plt.hist(array_bnnights)
plt.show()
```



Pandas

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

In [33]:
```
#CSV and display all the columns and their respective data types
df = pd.read_csv('hw_data.csv')
df.dtypes
```

Out[33]:
```
id          int64
sex        object
weight      int64
height      int64
dtype: object
```

2. Find the average weight

In [37]:
```
avg_weight = df["weight"].mean()
print("The average weight is: " + str(avg_weight))
```

```
The average weight is: 135.71428571428572
```

3. Find the Value Counts on column `sex`

In [38]:
```
df["sex"].value_counts()
```

Out[38]:
```
M    3
F    3
0    1
Name: sex, dtype: int64
```

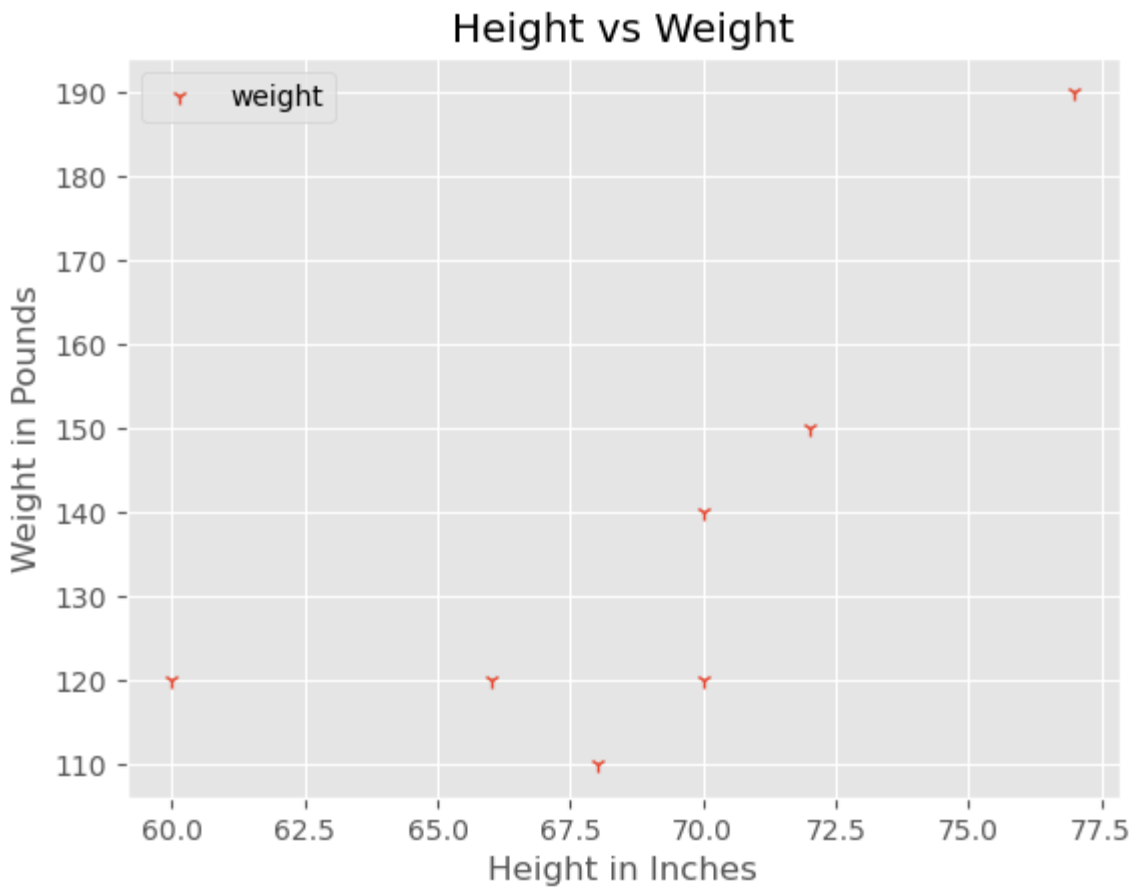
4. Plot Height vs. Weight

In [51]:
```
#not working for me in "style = "
plt.style.available
```

Out[51]:
```
['Solarize_Light2',
 '_classic_test_patch',
 '_mpl-gallery',
 '_mpl-gallery-nogrid',
 'bmh',
 'classic',
 'dark_background',
 'fast',
 'fivethirtyeight',
 'ggplot',
 'grayscale',
 'seaborn',
 'seaborn-bright',
 'seaborn-colorblind',
 'seaborn-dark',
 'seaborn-dark-palette',
 'seaborn-darkgrid',
 'seaborn-deep',
 'seaborn-muted',
 'seaborn-notebook',
 'seaborn-paper',
 'seaborn-pastel',
 'seaborn-poster',
 'seaborn-talk',
 'seaborn-ticks',
 'seaborn-white',
 'seaborn-whitegrid',
 'tableau-colorblind10']
```

In [54]:
```
df.plot("height", "weight", style = '1')
plt.title("Height vs Weight")
plt.xlabel("Height in Inches")
plt.ylabel("Weight in Pounds")
```

Out[54]: Text(0, 0.5, 'Weight in Pounds')



5. Calculate BMI and save as a new column

In [59]:
```
df["BMI"] = (df["weight"]*703/(df["height"]*df["height"]))
df
```

Out[59]:

	id	sex	weight	height	BMI
0	1	M	190	77	22.528251
1	2	F	120	70	17.216327
2	3	F	110	68	16.723616
3	4	M	150	72	20.341435
4	5	O	120	66	19.366391
5	6	M	120	60	23.433333
6	7	F	140	70	20.085714

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

In [60]:
```
df.to_csv("hw_dataB.csv")
```

Run the following (Mac)

In [61]:
```
!cat hw_dataB.csv
```

```
,id,sex,weight,height,BMI
0,1,M,190,77,22.528250969809413
1,2,F,120,70,17.216326530612246
2,3,F,110,68,16.723615916955016
3,4,M,150,72,20.341435185185187
4,5,O,120,66,19.366391184573004
5,6,M,120,60,23.433333333333334
6,7,F,140,70,20.085714285714285
```

Run the following (Windows)

In [ ]:
```
!type hw_dataB.csv
```