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]:
```
print ("helloworld")
```

```
helloworld
```

In [5]:
```
%%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.

In [14]:
```
def average(my_list):
    total = 0
    for item in my_list:
        total = total+item
    return total / len(my_list)

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

Out[14]:
```
3.375
```

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

In [17]:
```
def counts(my_list):
    counts = dict()
    for item in my_list:
        counts[item] = counts.get(item,0)+1
    return counts

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

Out[17]:
```
{1: 2, 2: 2, 4: 1, 3: 1, 5: 1, 9: 1}
```

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

In [227]:
```
import string
paragraph_text = '''
For a minute or two she stood looking at the house, and wondering what to do next, when suddenly a footman in l
ivery came running out of the wood—(she considered him to be a footman because he was in livery: otherwise, jud
ging by his face only, she would have called him a fish)—and rapped loudly at the door with his knuckles. It wa
s opened by another footman in livery, with a round face, and large eyes like a frog; and both footmen, Alice n
oticed, had powdered hair that curled all over their heads. She felt very curious to know what it was all abou
t, and crept a little way out of the wood to listen.
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 other, saying, in a solemn tone, 'For the Duchess. An invitation from the Queen to play croq
uet.' The Frog-Footman repeated, in the same solemn tone, only changing the order of the words a little, 'From
the Queen. An invitation for the Duchess to play croquet.'
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 s
he next peeped out the Fish-Footman was gone, and the other was sitting on the ground near the door, staring st
upidly up into the sky.
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 sa
me side of the door as you are; secondly, because they're making such a noise inside, no one could possibly hea
r you.' And certainly there was a most extraordinary noise going on within—a constant howling and sneezing, and
every now and then a great crash, as if a dish or kettle had been broken to pieces.
'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 instance, if you were inside, you might knock, and I could let you out, you know.' He was looki
ng up into the sky all the time he was speaking, and this Alice thought decidedly uncivil. 'But perhaps he can'
t help it,' she said to herself; 'his eyes are so very nearly at the top of his head. But at any rate he might
answer questions.—How am I to get in?' she repeated, aloud.
'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 hea
d: it just grazed his nose, and broke to pieces against one of the trees behind him.'''
words = paragraph_text.split()
from operator import itemgetter
N = 10
words2 = counts(words)
topnum = dict(sorted(words2.items(),key=itemgetter(1),reverse=True)[:N])
print("The ten words that appear the most are:", str(topnum))
```

```
The ten words that appear the most are: {'the': 32, 'and': 16, 'a': 15, 'to': 15, 'of': 9, 'was': 8, 'in': 7,
'she': 6, 'at': 6, 'he': 5}
```

4. Read in a file and write each line from the file to a new file Title-ized

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

Hint: There's a function to do this

In [215]:
```
txtfile = open("Homework2_textfile.txt")
lines = txtfile.readlines()
lines
```

Out[215]:
```
['My name is Jessica Coughlin.\n',
 'I am learning Python.\n',
 'I am working towards a new career as a data scientist.']
```

In [216]:
```
newfile = open("Homework2_newtext.txt","w+")
for word in lines:
    newfile.writelines(word.title())
newfile.close()
```

In [217]:
```
newfile2 = open("Homework2_newtext.txt")
lines2 = newfile2.readlines()
lines2
```

Out[217]:
```
['My Name Is Jessica Coughlin.\n',
 'I Am Learning Python.\n',
 'I Am Working Towards A New Career As A Data Scientist.']
```

Numpy

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

In [39]:
```
simple_list = [1,2,1,4,3,2,5,9]
np.mean(simple_list)
```

Out[39]:
```
3.375
```

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

In [91]:
```
heights = [174, 173, 173, 175, 171]
weights = [88, 83, 92, 74, 77]
#Turn heights list into numpy array while also converting cm to m.
heights_m = np.array(heights) / 100
print ("Heights in meters = ", heights_m)
#Turn weights list into numpy array.
weights_kg = np.array(weights)
print ("Weights in kg =", weights_kg)
np.set_printoptions(precision=2)
BMI = weights_kg / (heights_m * heights_m)
print ("The Body Mass Index (BMI) for each of these five individuals is", BMI)
```

```
Heights in meters =  [1.74 1.73 1.73 1.75 1.71]
Weights in kg = [88 83 92 74 77]
The Body Mass Index (BMI) for each of these five individuals is [29.07 27.73 30.74 24.16 26.33]
```

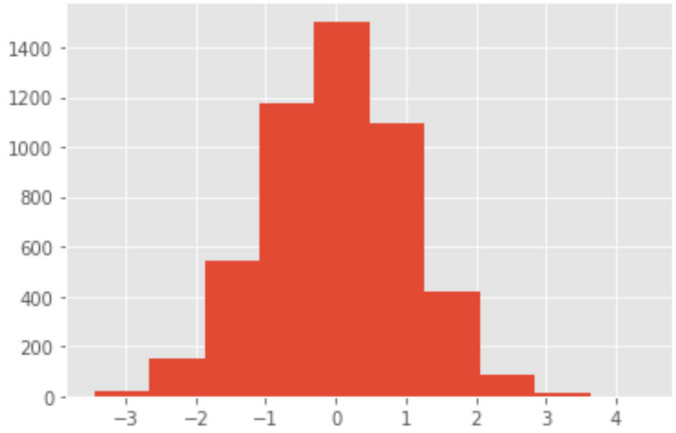
3. Create an array of length 20 filled with random values (between 0 to 1)

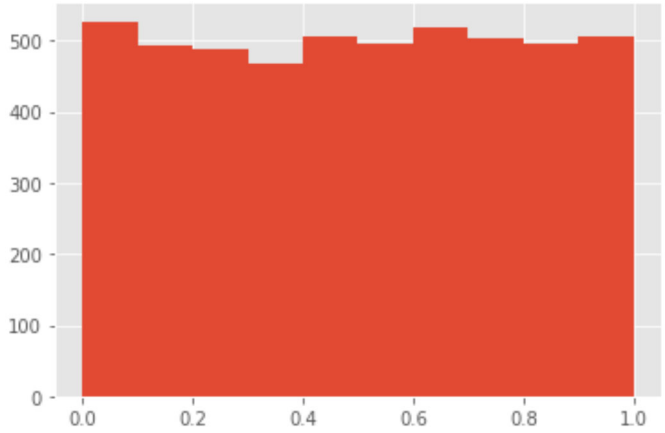
In [41]:
```
rand_twenty=np.random.rand(20)
print (rand_twenty)
```

```
[0.00407533 0.73860332 0.38969847 0.82107712 0.65066338 0.7678882
 0.56342866 0.56614446 0.3691496  0.67443251 0.06391834 0.19261668
 0.5014168  0.14595594 0.25177186 0.6276807  0.54838648 0.02290463
 0.70344011 0.96533826]
```

Bonus. 1. Create an array with a large (>1000) length filled with random numbers from different distributions (normal, uniform, etc.). 2. Then, plot a histogram of these values.

In [96]:
```
normal = np.random.randn(5000)
uniform = np.random.rand(5000)
plt.hist(normal)
plt.show()
plt.hist(uniform)
plt.show()
```





Pandas

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

In [115]:
```
df = pd.read_csv('C:/Users/bmpst/Desktop/JHU/Machine_Learning_and_Neural_Networks/mlnn_jess/2/hw_data.csv',index_col='id')
df.head()
```

Out[115]:

id	sex	weight	height
1	M	190	77
2	F	120	70
3	F	110	68
4	M	150	72
5	O	120	66

In [120]:
```
print ("The data types for each column are:\n", df.dtypes)
```

```
The data types for each column are:
 sex       object
weight     int64
height     int64
dtype: object
```

2. Find the average weight

In [140]:
```
print ("The average weight is:", round(df["weight"].mean(),2), "lbs")
```

```
The average weight is: 135.71 lbs
```

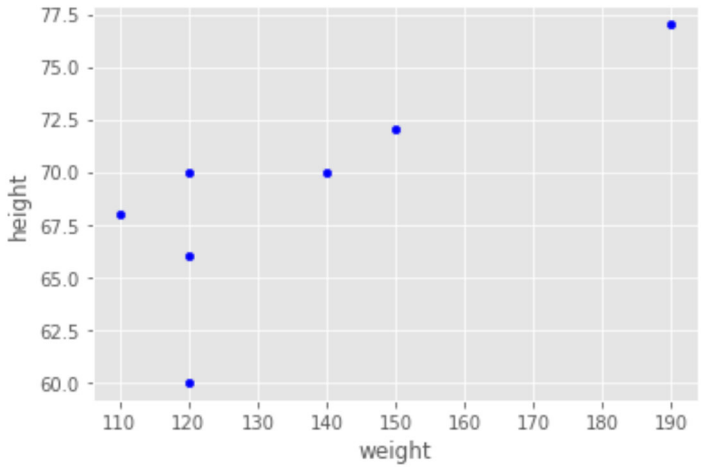
3. Find the Value Counts on column `sex`

In [144]:
```
print ("The number of entries for each item in the `sex` column are:", counts(df["sex"]))
```

```
The number of entries for each item in the `sex` column are: {'M': 3, 'F': 3, 'O': 1}
```

4. Plot Height vs. Weight

In [149]:
```
df.plot(kind = "scatter", x = "weight", y = "height", color = "blue")
plt.show()
```



5. Calculate BMI and save as a new column

In [151]:
```
df["BMI"] = df["weight"] / (df["height"] * df["height"]) *703
df.head()
```

Out[151]:

id	sex	weight	height	BMI
1	M	190	77	22.53
2	F	120	70	17.22
3	F	110	68	16.72
4	M	150	72	20.34
5	O	120	66	19.37

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

In [154]:
```
df.to_csv("hw_dataB.csv")
```

Run the following

In [155]:
```
!cat hw_dataB.csv
```

```
id,sex,weight,height,BMI
1,M,190,77,22.52825096980941
2,F,120,70,17.216326530612243
3,F,110,68,16.72361591695502
4,M,150,72,20.341435185185187
5,O,120,66,19.366391184573004
6,M,120,60,23.433333333333334
7,F,140,70,20.085714285714285
```