In [69]:

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

In [70]:

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

In [71]:

```
%%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 [72]:

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

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

Out[72]:

```
3.375
```

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

In [73]:

```
def counts(my_list):
    counts = dict()
    for item in my_list:
        counts[item] = my_list.count(item)

    return counts

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

Out[73]:

```
{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 [74]:

```
paragraph_text = '''
For a minute or two she stood looking at the house, and wondering what to do next, when
The Fish-Footman began by producing from under his arm a great letter, nearly as large a
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
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
‘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 h
‘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, straig

import string
counts(paragraph_text.translate(str.maketrans('','',string.punctuation)).lower().split()
```

Out[74]:

```
{'for': 5,
 'a': 15,
 'minute': 1,
 'or': 2,
 'two': 2,
 'she': 7,
 'stood': 1,
 'looking': 2,
 'at': 7,
 'the': 34,
 'house': 2,
 'and': 17,
 'wondering': 1,
 'what': 2,
 'to': 15,
 'do': 1,
 'next': 2,
 'when': 2,
 'suddenly': 1,
 'footman': 6,
 'in': 7,
 'livery': 3,
 'came': 2,
 'running': 1,
 'out': 5,
 'of': 9,
 'wood—she': 1,
 'considered': 1,
 'him': 3,
 'be': 2,
 'because': 3,
 'he': 6,
 'was': 8,
 'otherwise': 1,
 'judging': 1,
 'by': 3,
 'his': 5,
 'face': 2,
 'only': 2,
 'would': 1,
 'have': 1,
 'called': 1,
 'fish—and': 1,
 'rapped': 1,
 'loudly': 1,
 'door': 6,
 'with': 2,
 'knuckles': 1,
 'it': 3,
 'opened': 2,
 'another': 1,
 'round': 1,
 'large': 3,
 'eyes': 2,
 'like': 1,
 'frog': 1,
 'both': 2,
 'footmen': 1,
 'alice': 5,
 'noticed': 1,
 'had': 4,
 'powdered': 1,
 'hair': 1,
 'that': 3,
 'curled': 1,
 'all': 3,
 'over': 2,
 'their': 3,
 'heads': 1,
 'felt': 1,
 'very': 2,
 'curious': 1,
 'know': 1,
 'about': 1,
 'crept': 1,
 'little': 2,
 'way': 1,
 'wood': 2,
 'listen': 1,
 'fishfootman': 2,
 'began': 1,
 'producing': 1,
 'from': 2,
 'under': 1,
 'arm': 1,
 'great': 2,
 'letter': 1,
 'nearly': 2,
 'as': 4,
 'himself': 1,
 'this': 4,
 'handed': 1,
 'other': 2,
 'saying': 1,
 'solemn': 2,
 'tone': 2,
 '`for': 1,
 'duchess': 2,
 'an': 2,
 'invitation': 2,
 'queen': 2,
 'play': 2,
 'croquet’': 2,
 'frogfootman': 1,
 'repeated': 2,
 'same': 2,
 'changing': 1,
 'order': 1,
 'words': 1,
 '`from': 1,
 'then': 2,
 'they': 1,
 'bowed': 1,
 'low': 1,
 'curls': 1,
 'got': 1,
'entangled': 1,
'together': 1,
'laughed': 1,
'so': 2,
'much': 1,
'run': 1,
'back': 1,
'into': 3,
'fear': 1,
'hearing': 1,
'her': 2,
'peeped': 1,
'gone': 1,
'sitting': 1,
'on': 4,
'ground': 1,
'near': 1,
'staring': 1,
'stupidly': 1,
'up': 3,
'sky': 2,
'went': 2,
'timidly': 1,
'knocked': 1,
'‘there’s': 1,
'no': 2,
'sort': 1,
'use': 1,
'knocking’': 2,
'said': 3,
'‘and': 1,
'reasons': 1,
'first': 1,
'i’m': 1,
'side': 1,
'you': 5,
'are': 2,
'secondly': 1,
'they’re': 1,
'making': 1,
'such': 1,
'noise': 2,
'inside': 2,
'one': 2,
'could': 2,
'possibly': 1,
'hear': 1,
'you’': 1,
'certainly': 1,
'there': 1,
'most': 1,
'extraordinary': 1,
'going': 1,
'within—a': 1,
'constant': 1,
'howling': 1,
'sneezing': 1,
'every': 1,
'now': 1,
'crash': 1,
'if': 2,
'dish': 1,
'kettle': 1,
'been': 1,
'broken': 1,
'pieces': 2,
'‘please': 1,
'then’': 1,
'‘how': 1,
'am': 2,
'i': 3,
'get': 2,
'in’': 2,
'‘there': 1,
'might': 3,
'some': 1,
'sense': 1,
'your': 1,
'without': 1,
'attending': 1,
'‘if': 1,
'we': 1,
'between': 1,
'us': 1,
'instance': 1,
'were': 1,
'knock': 1,
'let': 1,
'know’': 1,
'time': 1,
'speaking': 1,
'thought': 1,
'decidedly': 1,
'uncivil': 1,
'‘but': 1,
'perhaps': 1,
'can’t': 1,
'help': 1,
'it’': 1,
'herself': 1,
'‘his': 1,
'top': 1,
'head': 2,
'but': 1,
'any': 1,
'rate': 1,
'answer': 1,
'questions—how': 1,
'aloud': 1,
'‘i': 1,
'shall': 1,
'sit': 1,
'here’': 1,
'remarked': 1,
'‘till': 1,
'tomorrow—’': 1,
'moment': 1,
'plate': 1,
'skimming': 1,
'straight': 1,
'footman’s': 1,
'just': 1,
'grazed': 1,
'nose': 1,
'broke': 1,
'against': 1,
'trees': 1,
'behind': 1}
```

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 [75]:

```
%%file text.txt
'One line of the file'
'Leads to another'
```

```
Overwriting text.txt
```

In [76]:

```
text = open('text.txt')
lines = text.read()
text.close()
title = lines.title()
with open('Title-ized.txt','w') as f:
    f.write(title)
```

Numpy

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

In [77]:

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

Out[77]:

```
3.375
```

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

In [78]:

```
heights = [174, 173, 173, 175, 171]
weights = [88, 83, 92, 74, 77]
heights_a = np.array(heights)
weights_a = np.array(weights)
(weights_a/heights_a/heights_a) * 10000
```

Out[78]:

```
array([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 [79]:

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

Out[79]:

```
array([0.38717578, 0.82348162, 0.80697456, 0.32039549, 0.55066091,
       0.97175279, 0.50546163, 0.59798669, 0.00815325, 0.33333825,
       0.57853619, 0.60426261, 0.66604183, 0.05614327, 0.86295934,
       0.14142788, 0.74994565, 0.51140315, 0.84697561, 0.1745057 ])
```

Bonus. 1. Create an array with a large (>1000) length filled with random numbers from different distributions (normal, uniform,

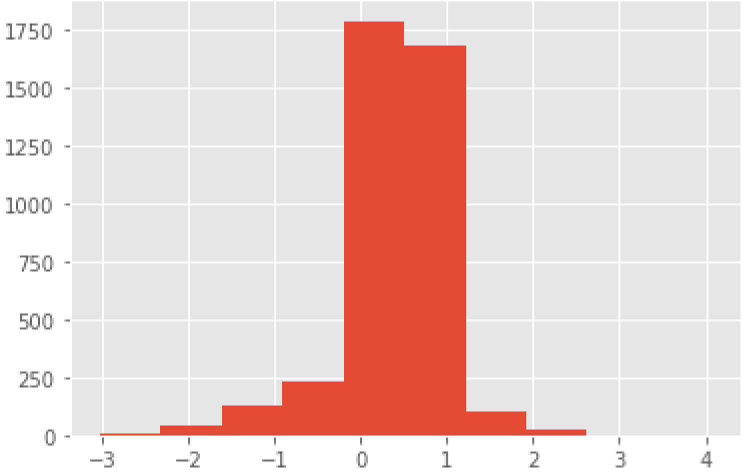
etc.). 2. Then, plot a histogram of these values.

In [80]:

```
normal = np.random.randn(1000)
uniform = np.random.rand(1000)
large_array = np.append(large_array,uniform)
plt.hist(large_array)
```

Out[80]:

```
(array([9.000e+00, 4.000e+01, 1.240e+02, 2.310e+02, 1.787e+03, 1.684e+03,
        9.800e+01, 2.400e+01, 2.000e+00, 1.000e+00]),
 array([-3.02084843, -2.31474502, -1.6086416 , -0.90253819, -0.19643477,
         0.50966864,  1.21577206,  1.92187547,  2.62797889,  3.3340823 ,
         4.04018572]),
 <BarContainer object of 10 artists>)
```



Pandas

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

In [81]:

```
df = pd.read_csv('hw_data.csv')
df.dtypes
```

Out[81]:

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

2. Find the average weight

In [82]:

```
df['weight'].mean()
```

Out[82]:

```
135.71428571428572
```

3. Find the Value Counts on column `sex`

In [83]:

```
df['sex'].value_counts()
```

Out[83]:

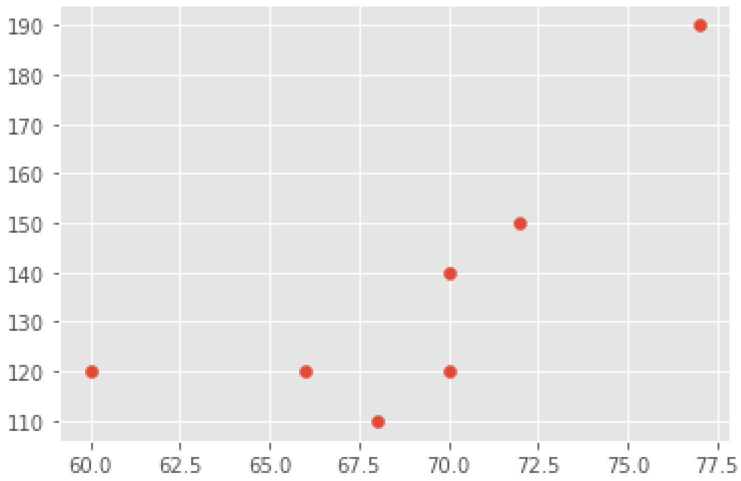
```
M    3
F    3
```

```
O    1
Name: sex, dtype: int64
```

4. Plot Height vs. Weight

```
In [84]: plt.scatter(df['height'],df['weight'])
```

```
Out[84]: <matplotlib.collections.PathCollection at 0x7fd3fc455e20>
```



5. Calculate BMI and save as a new column

```
In [85]: df['bmi'] = (df['weight']/df['height']**2)*703
         df['bmi']
```

```
Out[85]: 0    22.528251
         1    17.216327
         2    16.723616
         3    20.341435
         4    19.366391
         5    23.433333
         6    20.085714
         Name: bmi, dtype: float64
```

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

```
In [86]: df.to_csv('hw_dataB.csv')
```

Run the following

```
In [87]: !cat hw_dataB.csv
```

```
,id,sex,weight,height,bmi
0,1,M,190,77,22.52825096980941
1,2,F,120,70,17.216326530612243
2,3,F,110,68,16.72361591695502
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
```