

```
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.

```
In [82]: def average(my_list):
        total = 0
        for item in my_list:
            #do something with item!
            total = total + item
            avg = total / len(my_list)

        return avg

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

Out[82]: 3.375

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

```
In [102]: def counts(my_list):
           counts = dict()
           for item in my_list:
               #do something with item!
               if item in counts:
                   counts[item] += 1
               else:
                   counts[item] = 1

           return counts

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

```
Out[102]: {1: 2, 2: 2, 3: 1, 4: 1, 5: 1, 9: 1}
```

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

```
In [121]: paragraph_text = '''
For a minute or two she stood looking at the house, and wondering what to do.
The Fish-Footman began by producing from under his arm a great letter, nearly
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
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:—
'Please, then,' said Alice, 'how am I to get in?'
'There might be some sense in your knocking,' the Footman went on without attending
'I shall sit here,' the Footman remarked, 'till tomorrow—'
At this moment the door of the house opened, and a large plate came skimming
words = paragraph_text.split()
counts(words)
```

```
Out[121]: {'Alice': 4,
           'Alice.': 1,
           'An': 2,
           'And': 1,
           'At': 1,
           'But': 1,
           'Duchess': 1,
           'Duchess.': 1,
           'First': 1,
           'Fish-Footman': 2,
           'Footman': 2,
           'Footman.': 1,
           'Footman's': 1,
           'For': 2,
           'Frog-Footman': 1,
           'He': 1,
           'I': 3,
           'It': 1,
           'I'm': 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 [149]: text = open('hw_text.txt')
          lines = text.readlines()
          lines
```

```
Out[149]: ['This is the first line.\n',
           'This is the second line.\n',
           'This is the third line.']
```

```
In [158]: title_lines = []
          for x in lines:
              new_lines = x.title()
              title_lines.append(new_lines)
          title_lines
```

```
Out[158]: ['This Is The First Line.\n',
           'This Is The Second Line.\n',
           'This Is The Third Line.']
```

```
In [159]: with open('hw_titletext.txt', 'w') as out_file:
          out_file.write('\n'.join(title_lines))
```

```
In [160]: !cat hw_titletext.txt
```

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

```
Out[43]: 3.375
```

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

for-loop

```
In [46]: heights = [174, 173, 173, 175, 171]
weights = [88, 83, 92, 74, 77]
```

```
In [56]: heights_array = np.array(heights)
weights_array = np.array(weights)
```

```
In [69]: bmi = weights_array / (heights_array / 100)**2
bmi
```

```
Out[69]: array([ 29.06592681,  27.73229978,  30.73941662,  24.16326531,  26.332888
75])
```

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

```
In [45]: np.random.rand(20)
```

```
Out[45]: array([ 0.54814245,  0.78612592,  0.10785697,  0.01448447,  0.88968709,
 0.84530721,  0.77398996,  0.75257984,  0.9056188 ,  0.88968757,
 0.34301038,  0.3060278 ,  0.54255866,  0.80270063,  0.94392402,
 0.64044718,  0.15768503,  0.23740909,  0.39494757,  0.27571055])
```

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.

Pandas

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

```
In [8]: df = pd.read_csv('/Users/davidbradford/Documents/Johns Hopkins/Machine Learning/df')
```

Out[8]:

	sex	weight	height
id			
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

```
In [10]: df.columns
```

Out[10]: Index(['sex', 'weight', 'height'], dtype='object')

2. Find the average weight

```
In [16]: df.mean()
```

Out[16]: weight 135.714286
height 69.000000
dtype: float64

3. Find the Value Counts on column sex

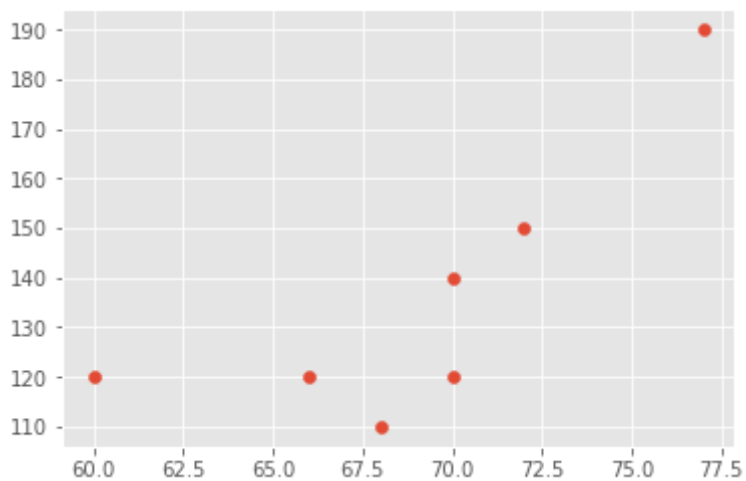
```
In [24]: sex_s = df.iloc[:,0]  
sex_s.value_counts()
```

Out[24]: M 3
F 3
O 1
Name: sex, dtype: int64

4. Plot Height vs. Weight

```
In [35]: plt.scatter(df['height'],df['weight'])
```

```
Out[35]: <matplotlib.collections.PathCollection at 0x110538cc0>
```



5. Calculate BMI and save as a new column

```
In [40]: weight = df['weight']  
height = df['height']  
df['BMI'] = (weight * .4536) / (height * .0254)**2  
df
```

```
Out[40]:
```

	sex	weight	height	BMI
id				
1	M	190	77	22.530860
2	F	120	70	17.218320
3	F	110	68	16.725552
4	M	150	72	20.343791
5	O	120	66	19.368634
6	M	120	60	23.436047
7	F	140	70	20.088040

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

```
In [41]: df.to_csv('hw_dataB.csv')
```

Run the following

```
In [42]: !cat hw_dataB.csv
```

```
id,sex,weight,height,BMI
1,M,190,77,22.53085970162495
2,F,120,70,17.218320150926015
3,F,110,68,16.725552482246837
4,M,150,72,20.34379068758138
5,O,120,66,19.368633778589874
6,M,120,60,23.436046872093744
7,F,140,70,20.08804017608035
```