

```
In [1]: import numpy as np
import pandas as pd
import scipy as sp
```

```
In [2]: %matplotlib inline
import matplotlib.pyplot as plt
plt.style.use('ggplot')
```

```
In [3]: %%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,M,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 [4]: def average(my_list):
        total = 0
        for item in my_list:
            total = total + item
        avg = total/len(my_list)
        return avg

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

Out[4]: 3.375

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

```
In [5]: 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[5]: {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 occurring words from the following paragraph. Bonus, remove punctuation from words.

```
In [6]: 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

# First remove punctuation from the string
punc = '!'()-[]{};:'"\,<>./?@$%^&*~'''
no_punct = ''
for char in paragraph_text:
    if char not in punc:
        no_punct = no_punct + char

# Split() function to create a list of words
cleaned_text = no_punct.split()

# Pass the list to the counts() function and sort by highest word count
sorted_list = sorted(counts(cleaned_text).items(), key=lambda x:x[1],r
sorted_dict = dict(sorted_list)

# Remove words with less than 3 occurrences
sorted_dict = dict(filter(lambda elem: elem[1] > 2, sorted_dict.items(
print(sorted_dict)

{'the': 32, 'and': 16, 'a': 15, 'to': 15, 'of': 9, 'was': 8, 'in': 7,
'she': 6, 'at': 6, 'door': 6, 'out': 5, 'he': 5, 'his': 5, 'Alice':
5, 'you': 5, 'had': 4, 'as': 4, 'this': 4, 'on': 4, 'footman': 3, 'li
very': 3, 'him': 3, 'because': 3, 'by': 3, 'large': 3, 'that': 3, 'al
l': 3, 'their': 3, 'for': 3, 'into': 3, 'up': 3, 'said': 3, 'Footman
': 3, 'I': 3, 'might': 3}
```

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 [7]: import os

original_file = open("mlnna2.txt", "r")
lines = original_file.readlines()

print("Writing the following lines to a new file:\n")
for line in lines:
    with open("newfile.txt", "a") as new_file:
        print(line.title())
        new_file.write(line.title())

original_file.close()
```

Writing the following lines to a new file:

This Is The First Line

My Name Is Elaine Burke

Once Upon A Time There Was A Little Girl

She Wrote A Program To Change The Text To Title Case.

The End

Numpy

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

```
In [8]: simple_list = [1,2,1,4,3,2,5,9]

np.mean(simple_list)
```

Out[8]: 3.375

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

```
In [9]: 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 [10]: np.random.rand(1,20)

Out[10]: array([[0.1328044 , 0.25201983, 0.16943245, 0.56465144, 0.01574666,
                0.23476072, 0.2590799 , 0.41953901, 0.99448688, 0.05806492,
                0.48917402, 0.38867479, 0.55048935, 0.55475855, 0.99603619,
                0.96511971, 0.01312108, 0.13541885, 0.58125308, 0.34657483]])
```

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 [11]: df=pd.read_csv('hw_data.csv')
df.dtypes

Out[11]: id          int64
sex            object
weight         int64
height         int64
dtype: object
```

2. Find the average weight

```
In [12]: df["weight"].mean()

Out[12]: 135.71428571428572
```

3. Find the Value Counts on column sex

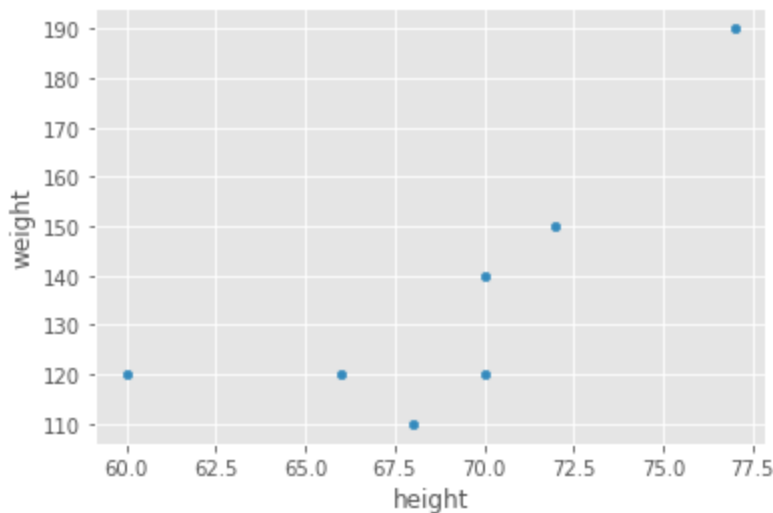
```
In [13]: df["sex"].value_counts()
```

```
Out[13]: M      3
         F      3
         0      1
         Name: sex, dtype: int64
```

4. Plot Height vs. Weight

```
In [14]: df.plot(x='height', y='weight', kind='scatter')
```

```
Out[14]: <AxesSubplot:xlabel='height', ylabel='weight'>
```



5. Calculate BMI and save as a new column

```
In [15]: # this assumes heights and weights need to be converted from inches an
df["bmi"] = round((df["weight"]/2.2) / ((df["height"]*.0254)**2),2)
df.head
```

```
Out[15]: <bound method NDFrame.head of      id sex  weight  height  bmi
0      1   M    190     77  22.58
1      2   F    120     70  17.25
2      3   F    110     68  16.76
3      4   M    150     72  20.39
4      5   0    120     66  19.41
5      6   M    120     60  23.48
6      7   F    140     70  20.13>
```

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

```
In [16]: df.to_csv('hw_dataB.csv')
```

Run the following

```
In [17]: !cat hw_dataB.csv  
  
      ,id,sex,weight,height,bmi  
0,1,M,190,77,22.58  
1,2,F,120,70,17.25  
2,3,F,110,68,16.76  
3,4,M,150,72,20.39  
4,5,O,120,66,19.41  
5,6,M,120,60,23.48  
6,7,F,140,70,20.13
```

```
In [ ]:
```