

```
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,O,120,66
6,M,120,60
7,F,140,70
```

Writing 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 [5]: 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[5]: 3.375

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

```
In [8]: 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[8]: {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 occurring words from the following paragraph. Bonus, remove punctuation from words.

```
In [32]: paragraph_text = '''
For a minute or two she stood looking at the house, and wondering what to do next,
The Fish-Footman began by producing from under his arm a great letter, nearly as la
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 t
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 out, s

import string
punct = string.punctuation + "-" + "\n" + "'" + '"'

paragraph_text1 = paragraph_text.translate(str.maketrans('', '', punct))
words = counts(paragraph_text1.split())

print(dict(sorted(words.items(), key=lambda item: item[1], reverse=True)))
```

```
{'the': 32, 'and': 17, 'a': 15, 'to': 15, 'of': 9, 'in': 8, 'was': 8, 'she': 6, 'at': 6, 'his': 6, 'door': 6, 'you': 6, 'out': 5, 'he': 5, 'had': 4, 'as': 4, 'this': 4, 'on': 4, 'For': 3, 'footman': 3, 'livery': 3, 'him': 3, 'because': 3, 'by': 3, 'large': 3, 'Alice': 3, 'that': 3, 'all': 3, 'their': 3, 'it': 3, 'for': 3, 'into': 3, 'up': 3, 'said': 3, 'Footman': 3, 'if': 3, 'I': 3, 'might': 3, 'or': 2, 'two': 2, 'looking': 2, 'house': 2, 'what': 2, 'next': 2, 'when': 2, 'came': 2, 'be': 2, 'face': 2, 'only': 2, 'with': 2, 'opened': 2, 'eyes': 2, 'both': 2, 'over': 2, 'very': 2, 'know': 2, 'little': 2, 'wood': 2, 'FishFootman': 2, 'from': 2, 'great': 2, 'nearly': 2, 'other': 2, 'solemn': 2, 'tone': 2, 'Duchess': 2, 'An': 2, 'invitation': 2, 'Queen': 2, 'play': 2, 'repeated': 2, 'same': 2, 'so': 2, 'her': 2, 'went': 2, 'no': 2, 'knocking': 2, 'are': 2, 'noise': 2, 'inside': 2, 'one': 2, 'could': 2, 'then': 2, 'am': 2, 'get': 2, 'But': 2, 'head': 2, 'minute': 1, 'stood': 1, 'wondering': 1, 'do': 1, 'suddenly': 1, 'running': 1, 'wood-she': 1, 'considered': 1, 'otherwise': 1, 'judging': 1, 'would': 1, 'have': 1, 'called': 1, 'fish-and': 1, 'rapped': 1, 'loudly': 1, 'knuckles': 1, 'It': 1, 'another': 1, 'round': 1, 'like': 1, 'frog': 1, 'footmen': 1, 'noticed': 1, 'powdered': 1, 'hair': 1, 'curled': 1, 'heads': 1, 'She': 1, 'felt': 1, 'curious': 1, 'about': 1, 'crept': 1, 'way': 1, 'listenThe': 1, 'began': 1, 'producing': 1, 'under': 1, 'arm': 1, 'letter': 1, 'himself': 1, 'handed': 1, 'saying': 1, 'croquet': 1, 'The': 1, 'FrogFootman': 1, 'changing': 1, 'order': 1, 'words': 1, 'From': 1, 'croquetThen': 1, 'they': 1, 'bowed': 1, 'low': 1, 'curls': 1, 'got': 1, 'entangled': 1, 'togetherAlice': 1, 'laughed': 1, 'much': 1, 'run': 1, 'back': 1, 'fear': 1, 'hearing': 1, 'peeped': 1, 'gone': 1, 'sitting': 1, 'ground': 1, 'near': 1, 'staring': 1, 'stupidly': 1, 'skyAlice': 1, 'timidly': 1, 'knockedThere': 1, 'sort': 1, 'use': 1, 'reasons': 1, 'First': 1, 'Im': 1, 'side': 1, 'secondly': 1, 'theyre': 1, 'making': 1, 'such': 1, 'possibly': 1, 'hear': 1, 'And': 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, 'dish': 1, 'kettle': 1, 'been': 1, 'broken': 1, 'piecesPlease': 1, 'how': 1, 'inThere': 1, 'some': 1, 'sense': 1, 'your': 1, 'without': 1, 'attending': 1, 'we': 1, 'between': 1, 'us': 1, 'instance': 1, 'were': 1, 'knock': 1, 'let': 1, 'He': 1, 'sky': 1, 'time': 1, 'speaking': 1, 'thought': 1, 'decidedly': 1, 'uncivil': 1, 'perhaps': 1, 'cant': 1, 'help': 1, 'herself': 1, 'top': 1, 'any': 1, 'rate': 1, 'answer': 1, 'questions-How': 1, 'aloudI': 1, 'shall': 1, 'sit': 1, 'here': 1, 'remarked': 1, 'till': 1, 'tomorrow-At': 1, 'moment': 1, 'plate': 1, 'skimming': 1, 'straight': 1, 'Footmans': 1, 'just': 1, 'grazed': 1, 'nose': 1, 'broke': 1, 'pieces': 1, 'against': 1, 'trees': 1, 'behind': 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 [38]: new_file = """With two on and one out in the bottom of the ninth, Andrew Benintendi

with open('file1.txt', 'w') as file1:
    file1.write(new_file)

with open("file1.txt", 'r') as my_file:
```

```

lines = my_file.readlines()

with open("output.txt", 'w') as outfile:
    for line in lines:
        outfile.write(line.title())

with open("output.txt", 'r') as output_file:
    print(output_file.read())

```

With Two On And One Out In The Bottom Of The Ninth, Andrew Benintendi Popped Up. Shortstop Gunnar Henderson Caught The Ball -- Umpires Had Called The Infield Fly Rule Anyway -- For The Second Out, Then Third-Base Umpire Junior Valentine Ruled That Andrew Vaughn Interfered With Henderson On The Play. "It Was A Big Surprise," Said Vaughn, Who Was The Runner At Second And Ruled Out. I Don'T Feel Like He Was Deterred From Making A Play. It Was A High Pop-Up. We Were All Reading It. As A Runner, I'Ve Got To Read It And Make Sure I Can Get Back To The Bag.

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

np.array(weights)/((np.array(heights)/100)**2)

```

Out[45]: 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 [46]: np.random.rand(20)

```

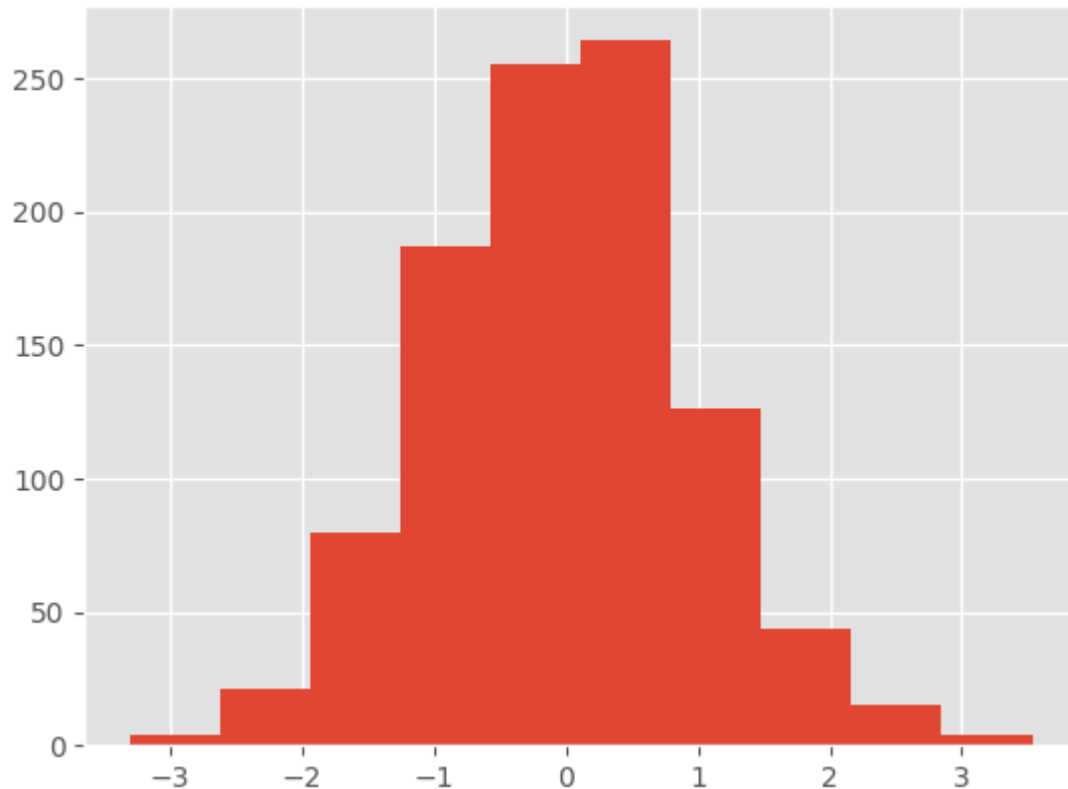
```
Out[46]: array([0.70741726, 0.74137658, 0.56522471, 0.63817476, 0.41428718,
                0.87207546, 0.13791462, 0.72272307, 0.38223821, 0.24856454,
                0.91562429, 0.12934952, 0.14071271, 0.92473567, 0.46337089,
                0.81968047, 0.74436974, 0.2684745 , 0.59141932, 0.0961094 ])
```

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

```
In [51]: rand = np.random.randn(1000)

plt.hist(rand)
```

```
Out[51]: (array([ 4., 21., 80., 187., 255., 264., 126., 44., 15., 4.]),
          array([-3.31010493, -2.62590214, -1.94169936, -1.25749658, -0.57329379,
                0.11090899, 0.79511177, 1.47931455, 2.16351734, 2.84772012,
                3.5319229 ]),
          <BarContainer object of 10 artists>)
```



Pandas

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

```
In [54]: df = pd.read_csv('hw_data.csv')
df.dtypes
```

```
Out[54]: id          int64
sex          object
weight       int64
height       int64
dtype: object
```

2. Find the average weight

```
In [62]: df['weight'].mean()
```

```
Out[62]: 135.71428571428572
```

3. Find the Value Counts on column sex

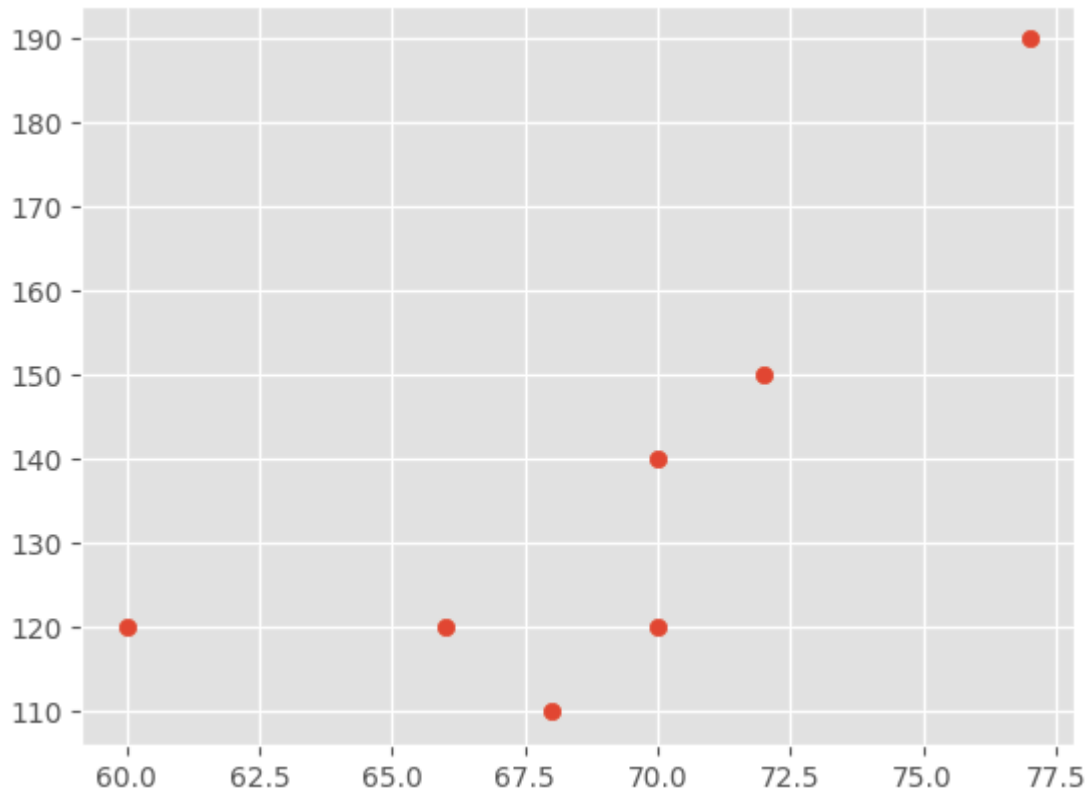
```
In [63]: df['sex'].value_counts()
```

```
Out[63]: sex
M      3
F      3
O      1
Name: count, dtype: int64
```

4. Plot Height vs. Weight

```
In [64]: plt.scatter(df['height'], df['weight'])
```

```
Out[64]: <matplotlib.collections.PathCollection at 0x2be6ba3a810>
```



5. Calculate BMI and save as a new column

```
In [68]: df['BMI'] = (df['weight'] / (df['height']**2))*703
```

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

```
In [69]: df.to_csv('hw_dataB.csv', index=False)
```

Run the following (Mac)

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

Run the following (Windows)

```
In [72]: !type 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
```

