In [326…

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

In [327…

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

In [328…

```
%%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 [329…

```
def average(my_list):
    x = sum(my_list) / len(my_list)
    return x

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

Out[329]:

```
3.375
```

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

In [330…

```
def counts(my_list):
    counts = dict()
    for i in my_list:
        counts[i] = counts.get(i, 0) + 1

    return counts

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

Out[330]:

```
{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 [331…

```
paragraph_text = '''
For a minute or two she stood looking at the house, and wondering what to do next, wh
The Fish-Footman began by producing from under his arm a great letter, nearly as larg
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 the
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. Fi
'Please, then,' said Alice, 'how am I to get in?'
'There might be some sense in your knocking,' the Footman went on without attending t
'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, str
```

In [332…

```
punc = '''!()-[]{};:'"\,<>./?@#$%^&*_~'''

for ele in paragraph_text:
    if ele in punc:
        paragraph_text = paragraph_text.replace(ele, "")

tot = counts(paragraph_text.split())
sorted(tot.items(), key = lambda x: x[1], reverse = True)
```

Out[332]:
```
[('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),
 ('livery', 3),
 ('him', 3),
 ('because', 3),
 ('by', 3),
 ('large', 3),
 ('that', 3),
 ('all', 3),
 ('their', 3),
 ('for', 3),
 ('into', 3),
 ('up', 3),
 ('said', 3),
 ('Footman', 3),
 ('I', 3),
 ('might', 3),
 ('For', 2),
 ('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),
 ('it', 2),
 ('little', 2),
 ('wood', 2),
 ('The', 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),
 ('croquet’', 2),
 ('repeated', 2),
 ('same', 2),
 ('so', 2),
 ('her', 2),
 ('sky', 2),
 ('went', 2),
 ('no', 2),
 ('knocking’', 2),
 ('are', 2),
 ('noise', 2),
 ('inside', 2),
 ('one', 2),
 ('could', 2),
 ('if', 2),
 ('pieces', 2),
 ('am', 2),
 ('get', 2),
 ('in’', 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),
 ('know', 1),
 ('about', 1),
 ('crept', 1),
 ('way', 1),
 ('listen', 1),
 ('began', 1),
 ('producing', 1),
 ('under', 1),
 ('arm', 1),
 ('letter', 1),
 ('himself', 1),
 ('handed', 1),
 ('saying', 1),
 ('‘For', 1),
 ('FrogFootman', 1),
 ('changing', 1),
 ('order', 1),
 ('words', 1),
 ('‘From', 1),
 ('Then', 1),
 ('they', 1),
 ('bowed', 1),
 ('low', 1),
 ('curls', 1),
 ('got', 1),
 ('entangled', 1),
 ('together', 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),
 ('timidly', 1),
 ('knocked', 1),
 ('‘There’s', 1),
 ('sort', 1),
 ('use', 1),
 ('‘and', 1),
 ('reasons', 1),
 ('First', 1),
 ('I’m', 1),
 ('side', 1),
 ('secondly', 1),
 ('they’re', 1),
 ('making', 1),
 ('such', 1),
 ('possibly', 1),
 ('hear', 1),
 ('you’', 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),
 ('then', 1),
 ('crash', 1),
 ('dish', 1),
 ('kettle', 1),
 ('been', 1),
 ('broken', 1),
 ('‘Please', 1),
 ('then’', 1),
 ('‘how', 1),
 ('‘There', 1),
 ('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),
 ('He', 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),
 ('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),
 ('At', 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 [333…

```
f = open("sample_text.txt", "r")
text = f.read()
text = text.title()
text
```

Out[333]:

```
'Here Is A Bunch Of Sample Text. Yay For Sample Text!'
```

In [334…

```
new = open("title-ized-text.txt", "w")
new.writelines(text)
```

Numpy

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

In [335…

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

np.mean(simple_list)
```

Out[335]:

```
3.375
```

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

In [336…

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

```
# Imperial BMI calculation
sq_heights = np.square(heights)
new_weights = np.array(weights)
new_weights = new_weights * 703
new_weights / sq_heights
```

Out[336]:
```
array([2.04, 1.95, 2.16, 1.7 , 1.85])
```

In [337…

```
# Metric BMI calculation (makes more sense based on the weight, but not height... pro
# I assume the numbers are in centimeters, so I multipled by 100 to convert the cm to
100 * (weights / sq_heights)
```

Out[337]:
```
array([0.29, 0.28, 0.31, 0.24, 0.26])
```

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

In [338…

```
array = np.random.rand(20,1)
array
```

Out[338]:
```
array([[0.83],
       [0.4 ],
       [0.99],
       [0.32],
       [0.37],
       [0.29],
       [0.07],
       [0.56],
       [0.54],
       [0.79],
       [0.14],
       [0.69],
       [0.75],
       [0.42],
       [0.14],
       [0.52],
       [0.78],
       [0.77],
       [0.36],
       [0.96]])
```

In [339…

```
len(array)
```

Out[339]:
```
20
```

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 [340…
```
# n/a
```

Pandas

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

In [341…
```
data = pd.read_csv("hw_data.csv")
data
```

Out[341]:

	id	sex	weight	height
0	1	M	190	77
1	2	F	120	70
2	3	F	110	68
3	4	M	150	72
4	5	O	120	66
5	6	M	120	60
6	7	F	140	70

2. Find the average weight

In [342…
```
weight = data["weight"]
np.mean(weight)
```

Out[342]:
```
135.71428571428572
```

3. Find the Value Counts on column `sex`

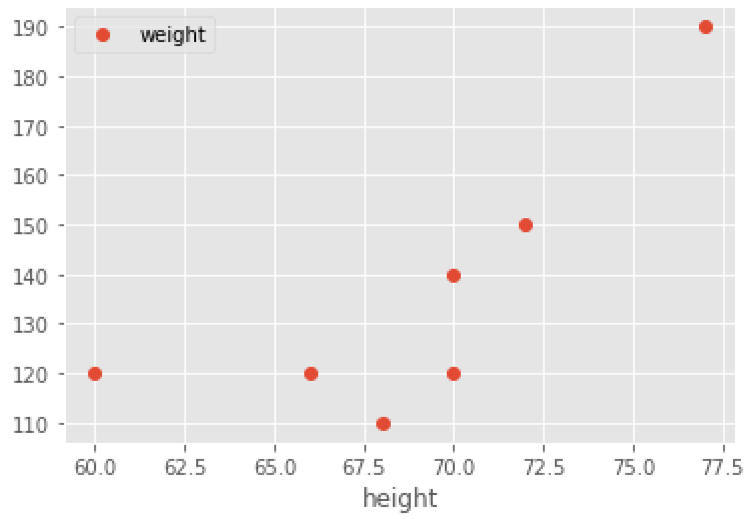
In [343…
```
sex = data["sex"]
pd.value_counts(sex)
```

Out[343]:
```
M    3
F    3
O    1
Name: sex, dtype: int64
```

4. Plot Height vs. Weight

In [344…
```
height = data["height"]
data.plot(x = "height", y = "weight", style = "o")
```

Out[344]:
```
<AxesSubplot:xlabel='height'>
```



5. Calculate BMI and save as a new column

In [345…

```
# This height/weight looks imperial...

bmi = (703 * weight) / (height ** 2)
data["bmi"] = bmi
data
```

Out[345]:

	id	sex	weight	height	bmi
0	1	M	190	77	22.528251
1	2	F	120	70	17.216327
2	3	F	110	68	16.723616
3	4	M	150	72	20.341435
4	5	O	120	66	19.366391
5	6	M	120	60	23.433333
6	7	F	140	70	20.085714

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

In [346…

```
data.to_csv("hw_dataB.csv")
```

Run the following

In [347…

```
!type hw_dataB.csv
```

```
,id,sex,weight,height,bmi
0,1,M,190,77,22.528250969809413
1,2,F,120,70,17.216326530612246
2,3,F,110,68,16.723615916955016
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
```

```
In [ ]:
```